

AD-A121 890 NON-NORMAL PROJECTILE PENETRATION IN SOIL AND ROCK.

1/2

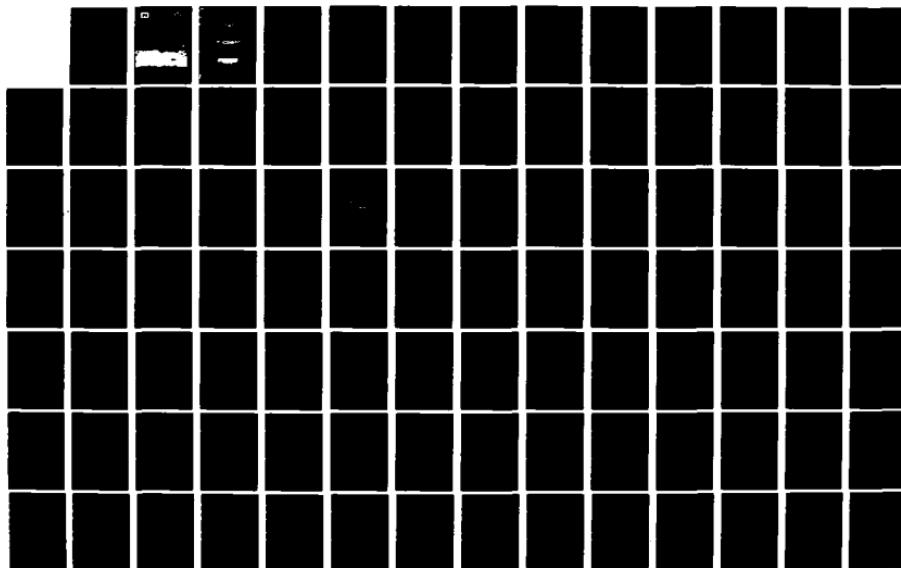
USER'S GUIDE FOR COMP. (U) ARMY ENGINEER WATERWAYS

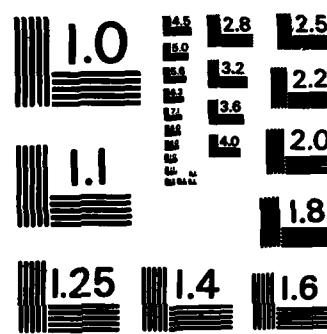
EXPERIMENT STATION VICKSBURG MS STRUC. D C CREIGHTON

UNCLASSIFIED SEP 82 WES/TR/SL-82-7

F/G 19/4

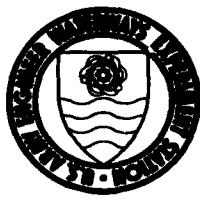
NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD A121800



12

TECHNICAL REPORT SL-82-7

NON-NORMAL PROJECTILE PENETRATION IN SOIL AND ROCK: USER'S GUIDE FOR COMPUTER CODE PENC02D

by

Daniel C. Creighton

Structures Laboratory

U. S. Army Engineer Waterways Experiment Station
P. O. Box 631, Vicksburg, Miss. 39180

September 1982

Final Report

DTIC
ELECTED
S NOV 29 1982
D
E

Approved For Public Release; Distribution Unlimited

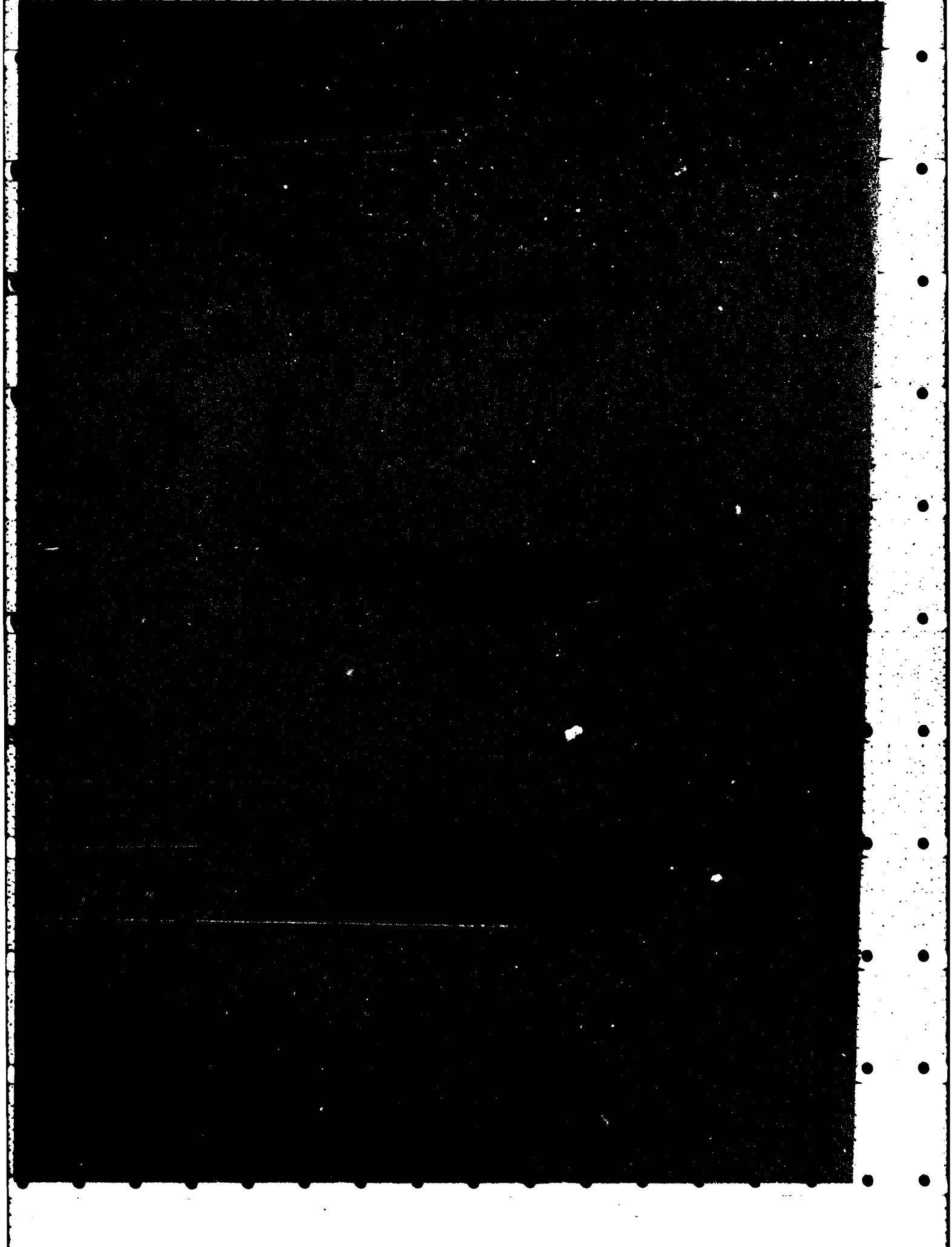


Prepared for Defense Nuclear Agency
Washington, D. C. 20305

Under DNA Task Y99QAXSC, Work Unit 00042,
"Earth Penetrator Support"

82 11 29 023

DATA FILE 0



Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER Technical Report SL-82-7	2. GOVT ACCESSION NO. AD 4121 890	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) NON-NORMAL PROJECTILE PENETRATION IN SOIL AND ROCK: USER'S GUIDE FOR COMPUTER CODE PENCO2D	5. TYPE OF REPORT & PERIOD COVERED Final report	
7. AUTHOR(s) Daniel C. Creighton	6. PERFORMING ORG. REPORT NUMBER	
9. PERFORMING ORGANIZATION NAME AND ADDRESS U. S. Army Engineer Waterways Experiment Station Structures Laboratory P. O. Box 631, Vicksburg, Miss. 39180	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Task Y99QAXSC, Work Unit 00042, "Earth Penetrator Support"	
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Nuclear Agency Washington, D. C. 20305	12. REPORT DATE September 1982	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	13. NUMBER OF PAGES 143	
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES Available from National Technical Information Service, 5285 Port Royal Road, Springfield, Va. 22151.		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) MOMENT (Computer program) PENCO2D (Computer program) Projectiles Soil penetration tests		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) A computer program called PENCO2D has been developed to analyze the non-normal impact and penetration of any rigid axisymmetric projectile into soil or rock. Horizontal target layering is allowed with accounts made for free-surface, interface, and wake separation-reattachment effects. Expressions for the compressive normal stress presented in U. S. Army Engineer Waterways Experiment Station Technical Report SL-79-15 are summed over the projectile surface, and the equations of motion are integrated step-by-step using (Continued)		

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

20. ABSTRACT (Continued)

a two-dimensional Newtonian integration scheme.

Variables and format of PENCO2D's input file, as well as output options, are given. To demonstrate use of PENCO2D, three sample problems are discussed, with setup and output shown.

A utility program called MOMENT, which calculates projectile weight, center-of-gravity location, and mass moments of inertia, is presented along with two example calculations.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

PREFACE

The work herein was sponsored by the Defense Nuclear Agency under Task Y99QAXSC, Work Unit 00042, "Earth Penetrator Support." The study was conducted by personnel of the Structures Laboratory (SL), U. S. Army Engineer Waterways Experiment Station (WES), during October 1981 through June 1982, under the general supervision of Mr. Bryant Mather, Chief, SL, and Dr. J. G. Jackson, Jr., Chief, Geomechanics Division (GD), SL. Mr. D. C. Creighton developed and implemented the computer analysis presented herein and prepared this report. Technical guidance was provided by Drs. Behzad Rohani and R. S. Bernard.

COL Tilford C. Creel, CE, was Commander and Director of WES during the course of this work. Mr. F. R. Brown was Technical Director.

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A	



CONTENTS

	<u>Page</u>
PREFACE	1
CONVERSION FACTORS, NON-SI TO SI (METRIC)	
UNITS OF MEASUREMENT	3
CHAPTER 1 INTRODUCTION	4
1.1 BACKGROUND	4
1.2 SCOPE	5
CHAPTER 2 USE OF PENC02D	6
2.1 BACKGROUND	6
2.2 INPUT	7
2.3 OUTPUT	8
2.4 SAMPLE PROBLEM 1: SIMULATION OF SANDIA TEST R 339-106 . .	10
2.5 SAMPLE PROBLEM 2: BLUNTED CONICAL PROJECTILE IN SOIL . .	13
2.6 SAMPLE PROBLEM 3: SIMULATION OF SANDSTONE 3-DEG REVERSE BALLISTIC TEST	17
TABLE 2.1	
FIGURES 2.1 THROUGH 2.12	
REFERENCES	68
APPENDIX A INTERFACE AND FREE-SURFACE EFFECTS	A1
FIGURE A.1	
APPENDIX B PENC02D'S NEWTONIAN INTEGRATION WITH GUIDELINES TO RELATED INPUT	B1
TABLE B.1	
FIGURES B.1 THROUGH B.3	
APPENDIX C GLOSSARY OF INPUT VARIABLES	C1
APPENDIX D LATERAL CUTOFF	D1
APPENDIX E USE OF ELAS TO SMOOTH NORMAL STRESS CURVE	E1
FIGURES E1 and E2	
APPENDIX F MOMENT: UTILITY PROGRAM TO CALCULATE WEIGHT, CENTER-OF-GRAVITY LOCATION, AND MOMENTS OF INERTIA FOR AXISYMMETRIC PROJECTILES	F1
FIGURES F.1 THROUGH F.10	
APPENDIX G LISTING OF PROGRAM PENC02D	G1
APPENDIX H NOTATION	H1

CONVERSION FACTORS, NON-SI TO SI (METRIC)
UNITS OF MEASUREMENT

Non-SI units of measurement used in this report can be converted to SI (metric) units as follows:

Multiply	By	To Obtain
g's (acceleration of gravity)	9.80665	metres per second squared
feet	0.3048	metres
feet per second	0.3048	metres per second
inches	0.0254	metres
inches per second	0.0254	metres per second
pounds (force)	4.44822	newtons
pounds (force) per square inch	6894.757	pascals
pounds (force)- seconds squared per inches fourth	10686893.0	kilograms per cubic metre
pound (mass)- inches squared	0.0002926397	kilogram-metres squared
pounds (mass) per cubic foot	16.01846	kilograms per cubic metre

NON-NORMAL PROJECTILE PENETRATION IN SOIL
AND ROCK: USER'S GUIDE FOR COMPUTER CODE PENCO2D

CHAPTER 1
INTRODUCTION

1.1 BACKGROUND

In the interest of developing a two-dimensional (2D) penetration code based on easily obtainable input, the U. S. Army Engineer Waterways Experiment Station (WES) acquired a copy of Avco Systems Division's differential-area force-law (DAFL) code from Picatinny Arsenal in 1975. The code, in its original form, uses a numerical predictor-corrector scheme to solve the nonlinear equations of motion for a rigid projectile in three dimensions (3D). The differential force acting on each surface element is specified as a function of the local surface velocity and a set of empirical target parameters. Avco's force law is intended to account for static resistance, surface friction, free-surface chipping, target layer interfaces, shock motion, and other phenomena in proportion to the values assigned to the various target parameters (Reference 1).

In the hands of an experienced user, the Avco DAFL code is a useful device for extrapolating laboratory penetration test results to field predictions. The various target parameters, however, are only vaguely related to standard engineering properties and are difficult or impossible to measure directly. For the engineer at large, the credibility of the code diminishes with the amount of guesswork needed for the target parameters.

From 1975-1979, WES undertook to redesign the DAFL code for users who are not necessarily penetration experts. The general structure of the code was changed in that motion in 3D was reduced to 2D, making the equations of motion linear and replacing the predictor-corrector scheme by simple Newtonian integration. The main thrust of the WES effort, however, was to replace Avco's differential force model with a model using standard properties and/or readily estimable empirical parameters.

With the WES 2D code, named PENCO2D, there is less need for guesswork on the part of the user. The major input parameters are unconfined strength for hard targets (rock, concrete, etc.) and S-number for soil (see Table I from Reference 2). The latter is a penetrability index that can usually be estimated within a factor of 2 from visual descriptions of the target.

1.2 SCOPE

This report presents PENCO2D as a black box; no attempt will be made to familiarize the reader with the detailed inner workings of the code. Instead, input and output variables (files and formats mainly) will be defined and discussed (Chapter 2). To demonstrate code use, three sample problems will be set up and run (also in Chapter 2), displaying the resulting input and output of PENCO2D. The theory has previously been presented (Reference 3); however, free surfaces above and below the penetrating projectile are handled by PENCO2D differently than in Reference 3, and are discussed in Appendix A. The 2D Newtonian integration scheme is outlined in Appendix B with specific regard to selection of time-step parameters. A glossary of input variables is given in Appendix C. PENCO2D monitors the instantaneous lateral motion throughout a given run and will cut it off completely if it becomes insignificant; guidelines for lateral-motion cutoff are discussed in Appendix D. Appendix E discusses an input parameter called ELAS, used to smooth the end of the stress distribution and stabilize the numerical scheme; guidelines are set forth to help the user select a reasonable value for ELAS. Appendix F introduces a utility program called MOMENT which can be used to calculate center-of-gravity (CG) location, total weight, and mass moments of inertia for a penetrator; these quantities are required input for PENCO2D. Appendix G gives a listing of PENCO2D, and Appendix H lists the notations used herein.

CHAPTER 2

USE OF PENCO2D

2.1 BACKGROUND

PENCO2D calculates the planar penetration of a rigid nonspinning axisymmetric projectile into any target which can be characterized using either S-number (soft soil-like materials) or density and yield strength (hard rock-like materials). Input includes projectile weight and geometry, target descriptors, initial conditions, separation and free-surface parameters, integration time-step definers, and variable limiters to determine problem end conditions. Output is available in both printed and plotted format. Printout includes an echo print of the input, calculated geometry table detailing each longitudinal element's shape and location, and step-by-step time history in both the projectile-fixed and target-fixed coordinate systems.

For the most part, PENCO2D retains the theory discussed in Reference 3. The methodology for handling free surfaces, however, has been modified somewhat; a discussion of the current treatment of layer interfaces and free surfaces appears in Appendix A. The original 3D version of Avco's DAFL code used a RUNGE-KUTTA integration for four time steps to start a predictor-corrector integration scheme used for the remainder of the penetration event. Once WES's version was cut to 2D, a simpler 2D Newtonian scheme replaced the original technique because numerical problems arose while trying to integrate the current normal-stress functional forms coupled with the often discontinuous separation and free-surface effects. Details of the current integration method in PENCO2D, including guidelines for selecting related input quantities, are given in Appendix B. It may be useful for a prospective user to first examine Appendices A through E, which all deal with PENCO2D input quantities and give some insight for selecting the more subjective input parameters.

2.2 INPUT

PENCO2D has one input file. Figure 2.1 is a record-by-record list of this file including the required format for each record. (Appendix C is a glossary of PENCO2D's input quantities and contains the definitions of all the variables appearing in Figure 2.1.) Each input data record is entirely alphanumeric, integer or real (i.e., no mixed-format records). The first record is the only alphanumeric record and contains an identifying title for the current problem in a left-justified 3X,A45 format; titles longer than 45 characters will be truncated when printed out. Integer records are read with a 12I6 format. Real records are read with either 4E18.6 or 5E14.4 format; Figure 2.1 gives details.

In addition to the input guidance detailed in Appendices A through E, Figures 2.2 and 2.3 will assist the user in the proper selection of some of the projectile's geometric parameters. Figure 2.2 details the four automatic nose geometries available in PENCO2D. If the projectile is of one of the types shown, the appropriate value of IOPSHP is input along with values for all of that shape's "required" parameters (indicated in Figure 2.2) and three other variables: DELS(1), NEMOV2, and ARATIO. With this information, PENCO2D automatically divides the projectile longitudinally into several elements. A typical longitudinal element is shown in Figure 2.3. DELS(1) defines the thickness of the first (nose-tip) element. NEMOV2 times 2 defines the number of circumferential elements into which each longitudinal element will be divided. In Figure 2.3, DA represents one of the surface differential areas that results after the circumferential division occurs, with L and W representing the length and width of each DA on that particular longitudinal element. Eliminating L and W from the expression for ARATIO in the figure and solving for DELS gives

$$\text{DELS}(i) = \frac{\text{ARATIO} \cdot \text{COS}(\text{THETA}(i)) \cdot \pi \cdot \text{R}(i)}{\text{NEMOV2}} \quad (2.1)$$

where the subscript *i* refers to the *i*th longitudinal element from the nose tip. This expression shows how changing ARATIO and NEMOV2 will

affect the sizes of the longitudinal elements. Increasing ARATIO and/or decreasing NEMOV2 produces fewer longitudinal elements, each having a larger thickness and larger differential areas.

If the current problem requires a nose shape different from the types shown in Figure 2.2, there is provision for an arbitrary nose shape initiated by inputting IOPSHP = 0. The R-, THETAD-, and DELS- arrays must be included at the end of the input file as shown on sheet 3 of Figure 2.1. NEL is the number of longitudinal elements to be input and is calculated automatically if IOPSHP > 0. The user need not input meaningful values of ARATIO and DELS(1) (sheet 1 of Figure 2.1) if an arbitrary shape is to be input, because neither quantity will be used.

Before concluding this section, it is worth noting that PENCO2D allows stacking of data sets in its input file. In fact, execution continues until an end-of-file is encountered in the input file. So, if the user has several problems to run, either separate input files can be set up and run or one large file with all the data sets sequentially listed can be run.

2.3 OUTPUT

PENCO2D generates both printed and plotted output. The printout is headed by the title that appears on the first line of the input file. How much output a user receives depends on four groups of input variables. The first group specifies how many plots (NUMPLTS), in addition to the two automatically-generated plots of projectile shape and final trajectory, and which plot types (ICHOOZ-array) will be output. The second group includes NPRINT, FREKOT, and FREQI and defines the output station frequency. The third group is NSTP, GAMSTP, YSTOP, ZSTOP, ALSTP, TIMEF, and VELF and encompasses the different ways to stop a problem. The fourth group includes NUMSTP, DGAMAD, and DALFAD and allows multiple executions with the same data by incrementing initial orientation and/or attack angle.

Plotting details are controlled with variables in the first group. If NUMPLTS = 0, the user gets two automatic plots. The first is a plot

of the resulting projectile based on the calculated R-, THETAD-, DELS-tables regardless of whether the user inputted the tables or allowed them to be generated automatically. The second plot gives a trajectory of the projectile's CG. If NUMPLTS > 0, then an array (ICHOOZ) must be input having NUMPLTS values. Each integer value of ICHOOZ indicates a desired plot type according to the table shown under the ICHOOZ listing in Appendix C. As shown in the table, NUMPLTS can be as large as 8 since there are eight types of additional plots available. If plot type "8" is to be selected, it must be the last member of the ICHOOZ-array.

The variables in group two are important because they determine how many output stations are to be generated. NPRINT allows the user to select whether the output locations will occur in equal time intervals (NPRINT = 1) or equal path-length intervals (NPRINT = 2). FREKOT is the size of the interval in seconds if NPRINT = 1, or in inches if NPRINT = 2. FREQI specifies how often, in projectile lengths, scaled projectiles appear on the automatic trajectory plot discussed above. Making FREQI equal to a very large number will result in projectiles plotted at the beginning and end of the trajectory only.

The variables in group three give the user several ways to complete the problem. The details of how each variable works are discussed under the appropriate listing in Appendix C. Each printed output station shows the current value of an output variable called RETURN. Table 2.1 lists the possible values of RETURN. All output stations except the last one should have RETURN = 0. The last output station always shows a value of RETURN greater than 1, indicating precisely why the problem ended. The user should check this to make sure the problem ended as intended.

The variables in group four allow re-use of the input data with a new projectile orientation and/or attack angle. The first run will set γ = GAMMAD degrees and α = ALPHAD degrees to begin the problem. After that problem is finished, γ and α are incremented so that the second run starts with γ = GAMMAD+DGAMAD degrees and α = ALPHAD+DALFAD degrees. In general, the i th run starts with

$$\gamma_i = \text{GAMMAD} + (i-1) \cdot \text{DGAMAD} \quad (2.2)$$

$$\alpha_i = \text{ALPHAD} + (i-1) \cdot \text{DALFAD} \quad (2.3)$$

where γ_i and α_i indicate initial values of γ and α , respectively, for the i th problem with the current data set. NUMSTP is the total number of runs to be made (i.e., i takes on values $1, 2, \dots, \text{NUMSTP}$). If only one run is desired with a data set, it is necessary to set $\text{NUMSTP} = 1$. A data set can be skipped over by making $\text{NUMSTP} \leq 0$.

The next three sections will demonstrate use of PENCO2D by example problems. Each problem will be verbally described, then the input file will be set up, and finally the resulting output will be shown.

2.4 SAMPLE PROBLEM 1: SIMULATION OF SANDIA TEST R 339-106

The first sample problem is a simulation of a test that was conducted by Sandia National Laboratories in May 1970 (Reference 4). Although PENCO2D was designed to handle 2D problems, this problem demonstrates how it can be used to solve a one-dimensional problem (i.e., normal impact).

Figure 2.4 depicts the projectile and initial conditions in the problem. The nose shape is an ogive; thus, the IOPSHP = 3 automatic generating routine can be used. From Figure 2.2 it is clear that only RJ (the ogive radius) and RO (the aftbody radius) need be specified, in addition to DELS(1), ARATIO, and NEMOV2 to automatically generate the longitudinal elements (see Section 2.1). Because this is a normal impact problem with no attack angle (i.e., $\gamma = 180$ deg, $\alpha = 0$), it is obvious that there should be no net lateral motion. To ensure this, the transverse moment of inertia was made very large (1.0×10^{20} lb-in 2)* to avoid possible rotation caused by computer round-off error. The aftbody of the actual projectile used in this test is nonflared and the

* A table of factors for converting non-SI units of measurement to SI (metric) units is presented on page 3.

motion is totally axial, so only tangential-type forces could occur on the aftbody. The theory in PENCO2D, however, contains no tangential force mechanisms (e.g., friction), and thus the aftbody can be made arbitrarily small. The location of the CG is also unimportant since there is no rotation.

The target is a three-layer system (1 foot of 3500-psi concrete over 1.5 foot of sand over a deep regime of cemented playa material). Because there is a hard layer over a softer layer, semifree-surface conditions exist at the concrete/sand interface (see Appendix A), and a reasonable estimate for I_T of the concrete should be included in the input. The density of the concrete is estimated to be 150 lb/ft^3 ($=0.2246 \times 10^{-3} \text{ lb-s}^2/\text{in}^4$). The S-numbers for the last two layers, sand and cemented playa, are estimated to be 10 and 1.75, respectively.

The projectile in the actual test penetrated about 162 inches. NPRINT was set to 2, so that path-length intervals are used for output incrementing. In order to obtain from 100 to 150 output increments, FREKOT was set at 1.25 inches (i.e., 162 inches/1.25 inches per increment \approx 130 increments). The only plot of interest, besides the automatic shape and trajectory plots, is the graph of axial acceleration versus path length (NUMPLTS = 1, ICHOOZ(1) = 8).

It is necessary to set only one stop condition within reasonable limits, that of final velocity. Ideally, one would like to run this problem until $V = 0$. However, at low velocities numerical problems arise that make it impossible to run all the way down to zero velocity. The smallest tolerable velocity is generally from 10 to 40 ft/s, depending on problem conditions. For this problem, 200 in/s ($=16.67 \text{ ft/s}$) will be used. Because the last layer is so hard, it is clear that the penetrator would not go much further after a speed of 16.67 ft/s is reached anyway; thus, it is a reasonable end to the problem. The other stop conditions are set so that they will not interfere with program execution.

The remainder of the input data is either straightforward or was achieved following guidelines outlined in Appendices A through E. The

resulting input file, when put together according to the format specified in Figure 2.1, looks like

SANDIA SLAB TEST R 339-106 (MAY 1970)						
8	3	195	4	1	1	60
8					0	2
0.250000E-01	0.200000E 01	0.100000E-01	0.			
0.	0.	0.740000E 02	0.			
0.	0.400000E 01	0.	0.			
0.300000E 02	0.250000E 02	0.240000E 02	0.			
0.	0.834000E 04	0.125000E 01	0.			
0.200000E 03	0.100000E 07	-0.100000E 07	0.			
0.200000E 00	0.100000E-02	0.100000E-06	0.100000E 00			
0.670000E 03	0.100000E 21	0.	0.500000E 01			
0.	0.100000E 01	0.250000E 01				
0.	0.200000E 03	0.	0.180000E 03			
0.	0.2246E-03	0.3500E 04	0.1200E 02	0.3500E 03		
1.0000E 01	0.	0.	0.3000E 02	0.1000E 02		
0.1750E 01	0.	0.	0.1000E 07	0.1000E 02		

Figure 2.5 shows the first three pages, a selected intermediate page, and the last page of the resulting printed output. As previously discussed, the first page echoes the input while the second page gives the resulting longitudinal sectioning of the projectile, as done by the automatic generating routine for IOPSHP = 3. The last element (I = 34) shows a THETAD-value close to -90 deg. The automatic generating routines add a thin (DELS = .001 inch) element to the tail end of the projectile with almost perpendicular sides to close the rear end. As long as the tail is closed, PENCO2D can handle situations in which the penetrator tumbles and/or flips over and goes backwards. Besides, this additional section does not affect forward motion at all. The column labeled S(I) is the axial distance from the center of each longitudinal element to the CG of the projectile (positive if forward of the CG). The column labeled RCAV(I) gives the local cylindrical wake-cavity radius r_c discussed in Section 2.4 of Reference 3 ("Wake Separation and Reattachment").

Sheet 3 of Figure 2.5 shows the first few output stations. Step 1 gives the initial conditions for the problem. Y and Z positions, velocities, forces, and g's refer to CG components in Y and Z (target-fixed) directions. Axial force and g's refer to net CG force and acceleration in the z (projectile-fixed axial) direction. All the

quantities labeled "PITCH" refer to rotational motion about the CG: ANGLE = γ , RATE = $\dot{\gamma}$, FORCE = $\int F_y dA$, g's = FORCE/m, MOMENT = $\int y dF_z - \int z dF_y$ (Equation B.3), where all integrals are taken over the entire projectile surface. DEL refers to the time step used just prior to outputting the current variable values. COUNT indicates how many time steps have elapsed since the last output station.

Sheet 4 of Figure 2.5 shows that lateral cutoff (Appendix D) occurs between steps 58 and 59, because the four rotational quantities PITCH RATE, PITCH FORCE, PITCH g's, and PITCH MOMENT have all been set to zero. PENCO2D does not allow cutoff to occur until the projectile is fully embedded in the deepest layer.

The last sheet of Figure 2.5 indicates that, as intended, there were about 130 output stations. The value of RETURN for the last step is 7, indicating that the CG velocity has dipped below the user-specified VELF (see Table 2.1), ending the problem as desired. PENCO2D also prints out final statistics consisting of total path length, elapsed time, final velocity, and peak values of lateral and axial accelerations.

Figure 2.6 gives the PENCO2D-generated plots for sample problem 1. The projectile shape curve (sheet 1) is provided for user verification. The trajectory plot (sheet 2) indicates the path traveled, with projectile orientation shown every FREQI projectile lengths. In this run, FREQI was input as 2.5. The horizontal lines extending across the trajectory plot indicate layer interfaces. Sheet 3 of Figure 2.6 is the requested axial deceleration versus path length graph (ICHO0Z = 8). This particular curve is useful because recorded test data are often plotted this way.

2.5 SAMPLE PROBLEM 2: BLUNTED CONICAL PROJECTILE IN SOIL

The second sample problem is an oblique impact problem to demonstrate the NUMSTP > 1 option to obtain multiple runs with the same data set. The multiple-run capability is to be used to search for the

limiting ricochet angle for a blunted conical-nosed projectile impacting a soil target ($S = 5$) at 750 ft/s.

The penetrator for this problem is shown in Figure F.3. Utility program MOMENT (Appendix F) was used to calculate weight (132.55 pounds), CG location (10.97 inches) from nose tip, and transverse moment of inertia (3793 lb-in²) as listed in Figure F.5. The shape can be program-generated using the IOPSHP = 4 option (Figure 2.2) with RB = 1.0 inch, RO = 3.0 inches, THETND = 21.8014 deg, and inputting appropriate values of ARATIO(-1.25) and NEMOV2(-10).

Because this example is concerned with projectile ricochet, a wake-separation angle $\phi_{\min} = 3$ deg (Reference 3) is to be used along with a free-surface parameter $I_T = 10$ (Appendix A). Wake separation and free-surface relief both contribute to the rotational motion necessary for ricochet; the values chosen for ϕ_{\min} and I_T are typical for a soil target.

An obliquity $\gamma = 230$ deg will be input with the attack angle $\alpha = 0$. Setting NUMSTP = 3 and DGAMAD = 15 deg produces three runs with initial values of $\gamma = 230, 245$, and 260 deg (Equation 2.2). Hopefully, $\gamma = 230$ deg will produce penetration without ricochetting, while $\gamma = 245$ deg or $\gamma = 260$ deg will produce ricochet. If this is the case, a limiting ricochet angle can be interpolated from these results; or another series can be executed, starting with the least oblique case that did not ricochet, and setting DGAMAD to a smaller increment. Plot types 2, 3, and 8 are to be selected (NUMPLTS = 3, ICHOOZ = 2,3,8) so that the effect of obliquity on lateral and axial accelerations can be examined.

The limiting variable again will be final velocity VELF (set = 200 in/s for this problem), as the purpose of the problem is to run the problem until the projectile either stops or ricochets. If it ricochets (i.e., exits the soil halfspace and reenters the air layer above it), PENCO2D will stop the calculation; the printout will indicate this end condition with a value of RETURN = 6 (see Table 2.1) in the last output station.

To select a printout increment, it is necessary to have at least a rough estimate of either the event's duration or total path length. An estimate of path length can be calculated using Young's penetration formula for normal impact (Equation 2 from Reference 2), which is

$$z_f = .0031 \cdot S \cdot N \cdot \sqrt{\frac{W}{A}} \cdot (V_o - 100) \quad , \quad V_o \geq 200 \text{ ft/s} \quad (2.4)$$

where A is the aftbody cross-sectional area ($=\pi r_o^2$). The units for this equation are W (pounds), $A(\text{in}^2)$, $V_o(\text{ft/s})$, and $z_f(\text{feet})$. Substituting $S = 5$, $W = 132.55$ pounds, $A = 28.27 \text{ in}^2$, and $V_o = 750 \text{ ft/s}$ (9000 in/s) and estimating the nose performance coefficient (see Table II from Reference 2) $N = 0.6$ for the blunted cone, Equation 2.4 gives $z_f = 13.09$ feet (157.1 inches). If NPRINT = 2 and FREKOT = 2.0 inches/output increment, then an estimate of the number of output stations is $z_f/\text{FREKOT} = 79$ (to the nearest whole number). If the user knows about how many output stations (N_{out}) are desired, then z_f/N_{out} will give an estimate of FREKOT.

The resulting PENCO2D input file looks like

BLUNTED CONE INTO S=5 SOIL (OBLIQUE IMPACT)							
10	4	198	2	3	3	20	0
2	3	8					2
0.			0.125000E 01		0.100000E-01		0.150000E 02
0.100000E 01			0.		0.		0.
0.			0.300000E 01		0.		0.
0.200000E 02			0.109700E 02		0.500000E 01		0.218014E 02
0.			0.900000E 04		0.200000E 01		0.
0.360000E 03			0.100000E 07		-0.100000E 07		0.
0.250000E 00			0.100000E-01		0.100000E-07		0.100000E 00
0.132550E 03			0.379300E 04		0.		0.300000E 02
0.			0.100000E 02		0.200000E 01		
0.300000E 01			0.200000E 03		0.		0.230000E 03
0.5000E 01		0.			0.1000E 07		0.1000E 02

Figure 2.7 lists pages 1, 2, 3, 10, 11, 18, 19, 22, and 23 from the 23-page printout. Sheet 2 lists the longitudinal elements that result from the selection of ARATIO = 1.25 and NEMOV2 = 10. The best way to obtain more elements is to lower ARATIO slightly. This is probably not necessary because ARATIO = 1.25 means that each DA is already close to

being square (ARATIO = 1.0 would be square). Thus, although there are only 23 elements, this is good enough for the current problem. The first element has a value of THETA almost equal to 90 deg. The reason for this thin (DELS = .001 inch) element is the same as for the rear element being almost -90 deg (explained for sample problem 1, Section 2.4). The projectile must be a closed surface in the direction of travel; the bluntness of the nose requires a near 90-deg element for closure.

Sheet 3 shows the beginning of the first run with step 1 showing the initial obliquity $\gamma = 230$ deg. Note that even at step 1 there are already finite values for all the forces and accelerations. PENC02D orients the projectile so that the nose tip sits initially at the origin of the target-fixed coordinate system, and this results in the initial partial embedment of one corner of the blunted nose. To resolve this problem, a user can input a negative value of ZSHIFT to start the calculation in the air layer (see Appendix C and sheet 1 of Figure B.2).

Sheet 4 shows the end of the first run with a value of RETURN = 7, indicating that the final velocity was reached. Thus, the projectile must still be embedded in the target as the final-position components indicate. There were 74 output stations for this problem which is close to the prerun estimate of 79.

Sheet 5 initiates the second run with the same data set with $\gamma = 245$ deg. The initial axial and pitch accelerations are both larger in magnitude than the case for $\gamma = 230$ deg because more of the nose is initially embedded as the angle of entry approaches the horizontal. Sheet 6 ends the second run, again indicating that the final velocity was reached (RETURN = 7) with the projectile still buried in the target.

Sheet 7 starts the last run of the sequence with $\gamma = 260$ deg. The projectile is nearly horizontal ($\gamma = 270$ is exactly horizontal). As expected, the initial axial and pitch accelerations are higher than the first two runs because a larger section of the nose is initially embedded.

The most interesting occurrence appears in step 37 (sheet 8), when all the forces and accelerations suddenly go to zero. The position

coordinates indicate that the projectile has ricocheted and reentered the air, but the calculation continues until the automatic check finally shuts down activities in step 46 (sheet 9). This is verified with the value of RETURN = 6.

Figure 2.8 shows the requested plots for this entire problem. Sheet 1 shows the shape graph, verifying the outer surface geometry of the projectile portrayed in Figure F.3. Sheets 2 through 5 represent output from the $\gamma = 230$ deg run, sheets 6 through 9 the $\gamma = 245$ deg run, and sheets 10 through 13 the $\gamma = 260$ deg run.

Comparing the three trajectory plots (sheets 2, 6, and 10), the increased initial embedment of the projectile with increasing initial γ is clearly evident, along with the ricochet for $\gamma = 260$ deg. Based on these three runs, the limiting ricochet angle (the smallest γ for ricochet) must be between 245 and 260 deg. The foregoing procedure could be repeated, starting with $\gamma = 245$ deg and incrementing by 5 deg, to more closely define the limiting angle.

Sheets 3, 7, and 11 are the three axial deceleration-time histories (ICHO0Z = 2) requested. Comparison of these three graphs shows a decreasing peak axial deceleration as γ increases. Similarly, sheets 4, 8, and 12 give the three lateral acceleration-time histories (ICHO0Z = 3) showing the negative extreme growing with increasing initial obliquity. As γ gets closer to horizontal, the projectile encounters greater lateral forces from below due to the top-side relief from the free surface. Because $\phi_{\min} > 0$, there is a cavity around the bottom side of the aftbody, and the unbalanced loads can easily cause rotation to occur. With the increased angle of attack, the axial forces must fall off more rapidly because these lateral loads are causing significant kinetic energy loss from the projectile.

2.6 SAMPLE PROBLEM 3: SIMULATION OF SANDSTONE 3-DEG REVERSE BALLISTIC TEST

In 1977, Sandia National Laboratories conducted four reverse ballistic tests (RBT) into sandstone for the Defense Nuclear Agency (DNA). Reference 5 contains a complete description of the test series.

The last sample problem is a simulation of two of the tests (Tests 2 and 3) which were done at a 3-deg angle of attack. Figure 2.9 indicates the nominal impact conditions for the tests: $V_0 = 1500 \text{ ft/s}$ (18,000 in/s), $\gamma = 183 \text{ deg}$, and $\alpha = 3 \text{ deg}$. Figure 2.10 shows the surface geometry of the actual test projectile. This sample calculation omits the aftbody flare (from diameter = 1.702 to 1.900 inches). The automatic generating routines can handle a flared aftbody if the diameters behind and in front of the flare section are the same. The RBT projectile, however, flares out to the new diameter (1.900 inches) and then holds that diameter constant. In the latter case, the user must input IOPSHP = 0 and input the entire R-, THETAD-, and DELS-arrays. For this example, IOPSHP = 3 will be used to show the automatic generating routine's ability to place the conical tip on the ogival nose (see Figure 2.2 for IOPSHP = 3). The flared aftbody will be replaced by a straight aftbody having a diameter of 1.702 inches.

Because of the dimensions of the test target, only the first 1.2 ms of the experiments were considered valid. Thus, the limiting variable for this problem will be TIMEF. If NPRINT = 1 (to set output increment units to time) and FREKOT = 0.00001 second, then there should be 122 ($=1+(TIMEF-TIMEI)/FREKOT$) output stations if TIMEI = 0 and TIMEF = 0.00121 second.

Of interest in this problem are axial deceleration-, lateral acceleration-, and total CG moment-time histories, because the projectiles in this test series had instrumentation to measure axial and lateral accelerations as well as axial strains at several locations. To obtain these plots, NUMPLTS has been set to 3 with the ICH00Z-array equal to 2, 3, and 4.

Because it is necessary to obtain more detail in this problem than in most penetration problems (i.e., FREKOT is small), the time-step parameters FRAD, FANG, NUMNOS (Appendix B) are much smaller than in the first two sample problems. They must be set small enough so that one or more integration steps occur between each printout step. In this problem it was decided to set the time-step parameters so that 7 to 11 integration steps would occur during each printout increment. For most

problems, the time-step parameters can be set much larger and still give a converged solution.

In any event, with all the above information taken into account, the following input file was used:

```
SANDSTONE RBT SLED TEST...ALPHA=3 DEG
 8   3   195   2   3   1   150   0   1
 2   3   4
 0.250000E-01   0.200000E 01   0.200000E-01   0.
 0.176000E 00   0.   0.102120E 02   0.
 0.   0.851000E 00   0.   0.
 0.181500E 02   0.871000E 01   0.379700E 01   0.
 0.   0.180000E 05   0.100000E-04   0.
 0.100000E 07   0.100000E 07   -0.100000E 07   0.
 0.200000E-01   0.300000E-03   0.200000E-07   0.200000E-04
 0.948000E 01   0.221900E 03   0.300000E 01   0.500000E 02
 0.   0.121000E-02   0.200000E 01   0.
 0.   0.240000E 03   0.   0.183000E 03
 0.   0.1947E-03   0.3400E 04   0.1000E 07   0.3500E 03
```

Figure 2.11 lists the first three pages and the last page of the resulting printout. Sheet 1 verifies the input discussed above. Sheet 2 reflects the longitudinal sectioning resulting from the IOPSHP = 3 shape routine. The conical nose tip is evident, as the first six elements all show a constant surface angle THETAD of 50.3 deg. Sheet 3 is the beginning of the problem. Step 1 shows a small initial Y-velocity ($.5 \times 10^{-3}$ in/s) when it should be identically equal to zero, indicating the presence of computer round-off error. The last sheet is the end of the problem (time equal to TIMEF). RETURN = 3 in step 122 verifies that TIMEF was the source of problem termination.

Figure 2.12 exhibits the requested plots for this problem. Sheets 1 and 2 automatically verify the projectile shape and path traveled. Sheets 3, 4, and 5 show the axial deceleration-, lateral acceleration-, and CG moment-time histories, respectively.

Table 2.1 Values of RETURN in PENCO2D printout.

<u>RETURN</u>	<u>EXPLANATION</u>
0	Code executing normally; no stop conditions encountered yet.
2	Maximum number of printout increments (NSTP) has been reached; calculation stops.
3	Current time t has exceeded TIMEF; calculation stops.
4	Absolute value of Y (horizontal coordinate of CG) has exceeded YSTOP; calculation stops.
5	Value of Z (vertical coordinate of CG) has become less than ZSTOP; calculation stops.
6	Projectile has exited the earth target and passed up into the air (layer 1); calculation stops.
7	CG velocity V has become less than VELF; calculation stops.
8	Orientation angle γ has exceeded GAMSTP; calculation stops.
9	Absolute value of angle of attack α has exceeded ALSTP; calculation stops.

TITLE									
3X,A45									

NEMOV2	IOPSHP	NSTP	NLAY	NUMPLTS	NUMSTP	NUMNOS	IREDUC	NPRINT	
I6	I6	I6	I6	I6	I6	I6	I6	I6	

Note: Include this record only if NUMPLTS > 0.

ICHO0Z(1)	ICHO0Z(2)	ICHO0Z (NUMPLTS)	
I6	I6	I6	I6	I6	I6	I6	I6	

DELS(1)	ARATIO	ELAS	DGAMAD
E18.6	E18.6	E18.6	E18.6

RB	EJ	RJ	RN
E18.6	E18.6	E18.6	E18.6

RC	RQ	XFA	XFF
E18.6	E18.6	E18.6	E18.6

XLP	XCG	SN	THETND
E18.6	E18.6	E18.6	E18.6

THETFD	VEL	FREKOT	DALFAD
E18.6	E18.6	E18.6	E18.6

Figure 2.1 Data format for input file 5 (Sheet 1 of 3).

GAMSTP	YSTOP	ZSTOP	ZSHIFT
E18.6	E18.6	E18.6	E18.6

FRAD	FANG	DTMIN	DTMAX
E18.6	E18.6	E18.6	E18.6

WEIGHT	XICG	ALPHAD	ALSTP
E18.6	E18.6	E18.6	E18.6

TIMEI	TIMEF	FREQI	
E18.6	E18.6	E18.6	

PHIMIN	VELF	W1I	GAMMAD
E18.6	E18.6	E18.6	E18.6

Note: Include this record for each target layer I.
 $2 \leq I \leq NLAY$ (Layer 1 is air and need not be specified)

SNUM(I)	DENSITY(I)	YIELD(I)	ZM(I)	XIRD(I)
E14.4	E14.4	E14.4	E14.4	E14.4

End here if IOPSHP $\neq 0$.

Note: Include the following records only if IOPSHP = 0.

NEL	
16	

Figure 2.1 (Sheet 2 of 3).

Note: Include enough records to exhaust entire THETAD-array.

THETAD(1)	THETAD(2)	...	THETAD(NEL)
E18.6	E18.6	E18.6	E18.6

Note: Include enough records to exhaust entire R-array.

R(1)	R(2)	...	R(NEL)
E18.6	E18.6	E18.6	E18.6

Note: Include enough records to exhaust entire DELS-array.

DELS(1)	DELS(2)	...	DELS(NEL)
E18.6	E18.6	E18.6	E18.6

Figure 2.1 (Sheet 3 of 3).

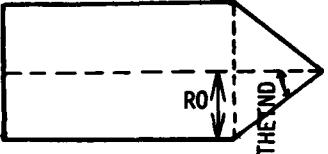
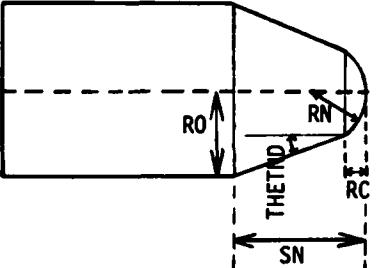
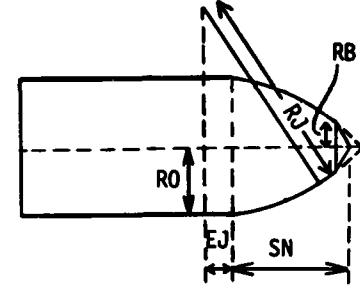
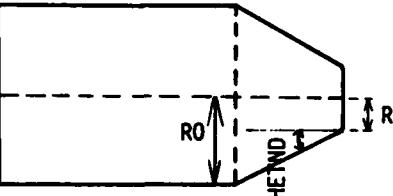
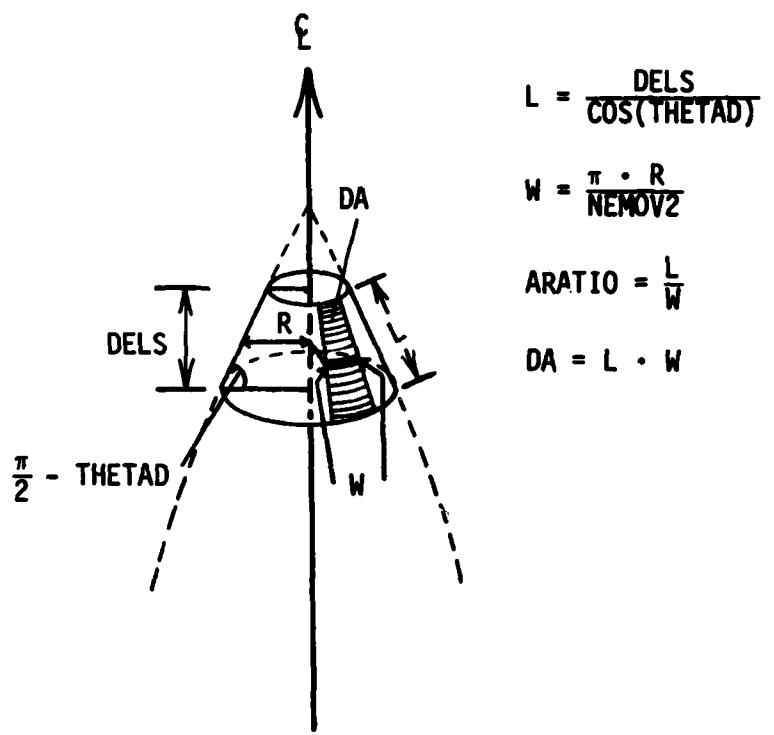
Value of IOPSHP	Shapes Showing Required Input	Remarks
1		SN (nose length) may be input instead of THETEND; THETEND will then be calculated; if THETEND is input, SN will be calculated.
2		All input shown is required.
3		(a) If a conical tip is desired, all input shown is required. (b) If no conical tip is desired, set RB = 0; input either SN or RJ and the other will be calculated; all other input shown is required.
4		All input shown is required; SN is calculated; DELS(1) is not required input.

Figure 2.2 PENCO2D projectile shape options.



$$L = \frac{DELS}{\cos(\text{THETAD})}$$

$$W = \frac{\pi \cdot R}{\text{NEMOV2}}$$

$$\text{ARATIO} = \frac{L}{W}$$

$$DA = L \cdot W$$

Figure 2.3 Longitudinal element.

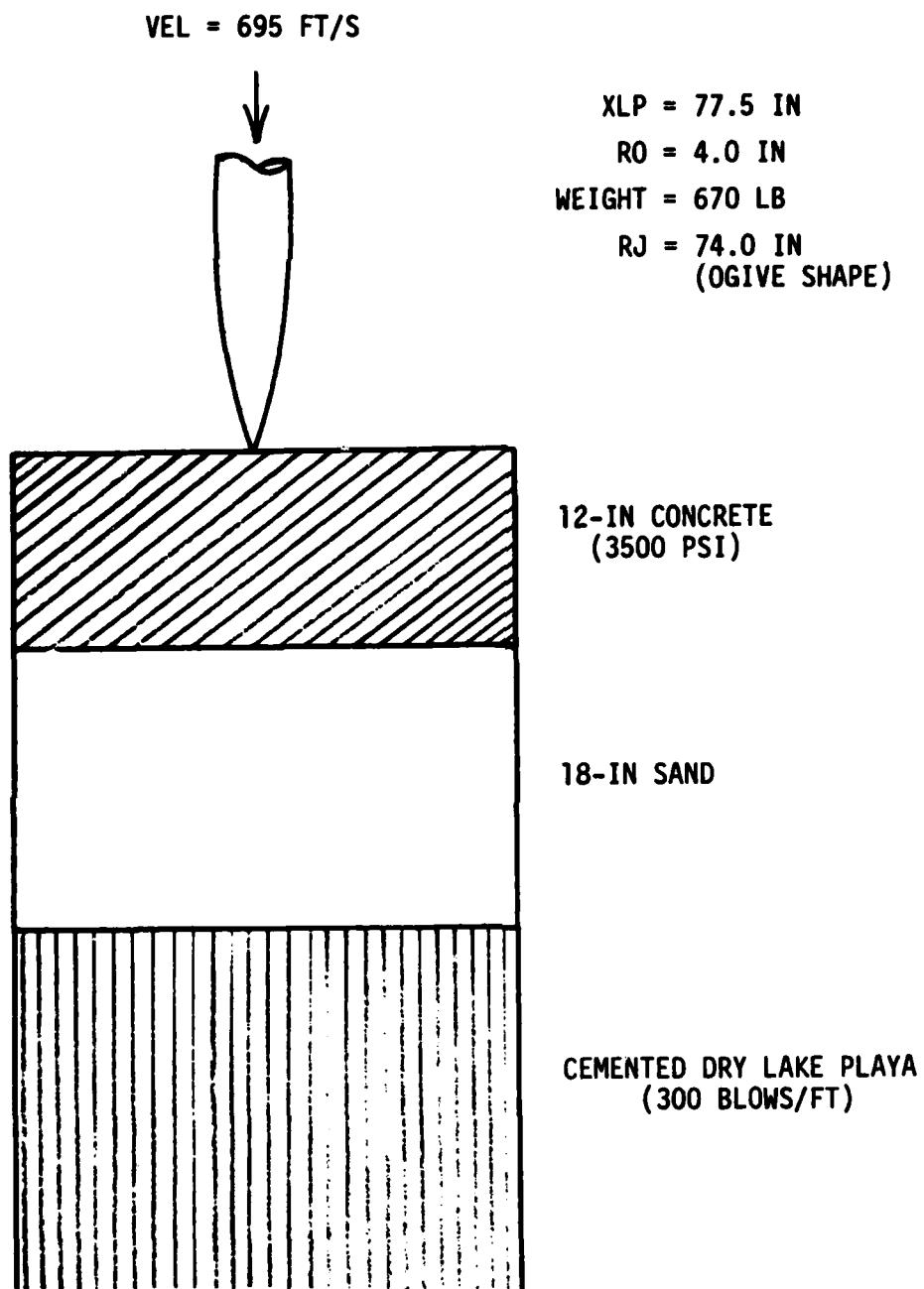


Figure 2.4 Initial conditions for sample problem 1.

SANDIA SLAB TEST R 339-106 (MAY 1970)

```

*****  

* 2D NEWTONIAN PENETRATION CODE *  

*****  

NENG2= 6 10PSP= 3 NSTP= 195 NLAY= 4 NMPLT= 1 NSTPS= 1 NMUS= 60 IREDAC= 0 NPRINT= 2
ADDITIONAL PLOTS... 0
DELS(1)= 0.250000E-01 ARATIO= 0.200000E 01 ELAS= 0.100000E-01 DCAMAD= 0.
RB= 0. EJ= 0. RJ= 0.740000E 02 RN= 0.
RC= 0. RG= 0.400000E 01 XFA= 0. XFR= 0.
XLP= 0.300000E 02 XCG= 0.250000E 02 SH= 0.240000E 02 THETAD= 0.
THETFD= 0. VEL= 0.634000E 04 FREKOT= 0.125000E 01 DALLAD= 0.
GAMSTP= 0.200000E 03 YSTOP= 0.100000E 07 ZSTOP= 0.100000E 07 ZSWIT= 0.
FRAD= 0.200000E 00 FANG= 0.100000E-02 DTHIN= 0.100000E-06 DTHMAX= 0.100000E 00
WEIGHT= 0.670000E 03 XICG= 0.100000E 21 ALPHAD= 0.
TIMEF= 0.100000E 01 FREQT= 0.250000E 01
PHMIN= 0. DEG. VELF= 0.200000E 03 WIJ= 0. GAMMAD= 0.100000E 03
LAYER NO. (YOUNG'S NUMBER) (LB-DENSITY) (PSI) (LAYER BDEPTH-IN.) (TENS. KIP/inch)
1 *** AIR *** NO RESISTANCE ***  

2 0. 2.2460E-04 3.500E 03 1.200E 01 350.0
3 10.00 0. 0. 3.000E 01 10.0
4 1.75 0. 0. 1.000E 00 10.0

```

Figure 2.5 Selected pages from the printed output for sample problem 1 (Sheet 1 of 5).

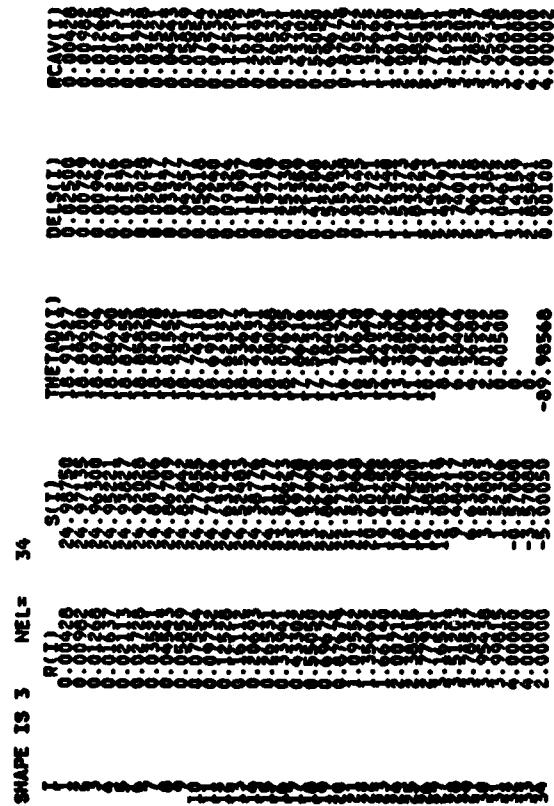


Figure 2.5 (Sheet 2 of 5).

Figure 2.5 (Sheet 3 of 5).

Figure 2.5 (Sheet 4 of 5).

```

STEP = 122  Y-POSITION = 0.2699E-01  COUNT = 1
TIME = 0.2699E-01  Y-VELOCITY = 0.160000E-03  COUNT = 1
X-FORCE = 0.1755E-02  X-VELOCITY = 0.1755E-03
X-AXIAL FORCE = 0.1755E-02  X-AXIAL FORCE = 0.1755E-02
STEP = 123  Y-POSITION = 0.2946E-01  COUNT = 1
TIME = 0.2946E-01  Y-VELOCITY = 0.160000E-03  COUNT = 1
Y-AXIAL G'S = 0.1755E-02  Y-AXIAL G'S = 0.1755E-02
X-AXIAL FORCE = 0.1755E-02  X-AXIAL FORCE = 0.1755E-02
STEP = 124  Y-POSITION = 0.2988E-01  COUNT = 1
TIME = 0.2988E-01  Y-VELOCITY = 0.160000E-03  COUNT = 1
Y-AXIAL G'S = 0.1755E-02  Y-AXIAL G'S = 0.1755E-02
X-AXIAL FORCE = 0.1755E-02  X-AXIAL FORCE = 0.1755E-02
STEP = 125  Y-POSITION = 0.3076E-01  COUNT = 1
TIME = 0.3076E-01  Y-VELOCITY = 0.160000E-03  COUNT = 1
Y-AXIAL G'S = 0.1755E-02  Y-AXIAL G'S = 0.1755E-02
X-AXIAL FORCE = 0.1755E-02  X-AXIAL FORCE = 0.1755E-02
STEP = 126  Y-POSITION = 0.3174E-01  COUNT = 1
TIME = 0.3174E-01  Y-VELOCITY = 0.160000E-03  COUNT = 1
Y-AXIAL G'S = 0.1755E-02  Y-AXIAL G'S = 0.1755E-02
X-AXIAL FORCE = 0.1755E-02  X-AXIAL FORCE = 0.1755E-02
STEP = 127  Y-POSITION = 0.3239E-01  COUNT = 1
TIME = 0.3239E-01  Y-VELOCITY = 0.160000E-03  COUNT = 1
Y-AXIAL G'S = 0.1755E-02  Y-AXIAL G'S = 0.1755E-02
X-AXIAL FORCE = 0.1755E-02  X-AXIAL FORCE = 0.1755E-02
STEP = 128  Y-POSITION = 0.3351E-01  COUNT = 1
TIME = 0.3351E-01  Y-VELOCITY = 0.160000E-03  COUNT = 1
Y-AXIAL G'S = 0.1755E-02  Y-AXIAL G'S = 0.1755E-02
X-AXIAL FORCE = 0.1755E-02  X-AXIAL FORCE = 0.1755E-02
STEP = 129  Y-POSITION = 0.3499E-01  COUNT = 1
TIME = 0.3499E-01  Y-VELOCITY = 0.160000E-03  COUNT = 1
Y-AXIAL G'S = 0.1755E-02  Y-AXIAL G'S = 0.1755E-02
X-AXIAL FORCE = 0.1755E-02  X-AXIAL FORCE = 0.1755E-02
STEP = 130  Y-POSITION = 0.3700E-01  COUNT = 1
TIME = 0.3700E-01  Y-VELOCITY = 0.160000E-03  COUNT = 1
Y-AXIAL G'S = 0.1755E-02  Y-AXIAL G'S = 0.1755E-02
X-AXIAL FORCE = 0.1755E-02  X-AXIAL FORCE = 0.1755E-02
STEP = 131  Y-POSITION = 0.3902E-01  COUNT = 1
TIME = 0.3902E-01  Y-VELOCITY = 0.160000E-03  COUNT = 1
Y-AXIAL G'S = 0.1755E-02  Y-AXIAL G'S = 0.1755E-02
X-AXIAL FORCE = 0.1755E-02  X-AXIAL FORCE = 0.1755E-02

```

```

FINAL STATISTICS
PATH LENGTH TRAVELED = 0.16211E-03 IN. ( 0.13509E-02 FT.)
TIME ELAPSED = 0.15910E-02 SEC
FINAL SPEED = 0.16441E-03 FT/SEC
FINAL LATERAL ACCELERATION = 0.11172E-03 G'S
MAX. LATERAL ACCELERATION = 0.11253E-03 G'S

```

Figure 2.5 (Sheet 5 of 5).

PROJECTILE SHAPE

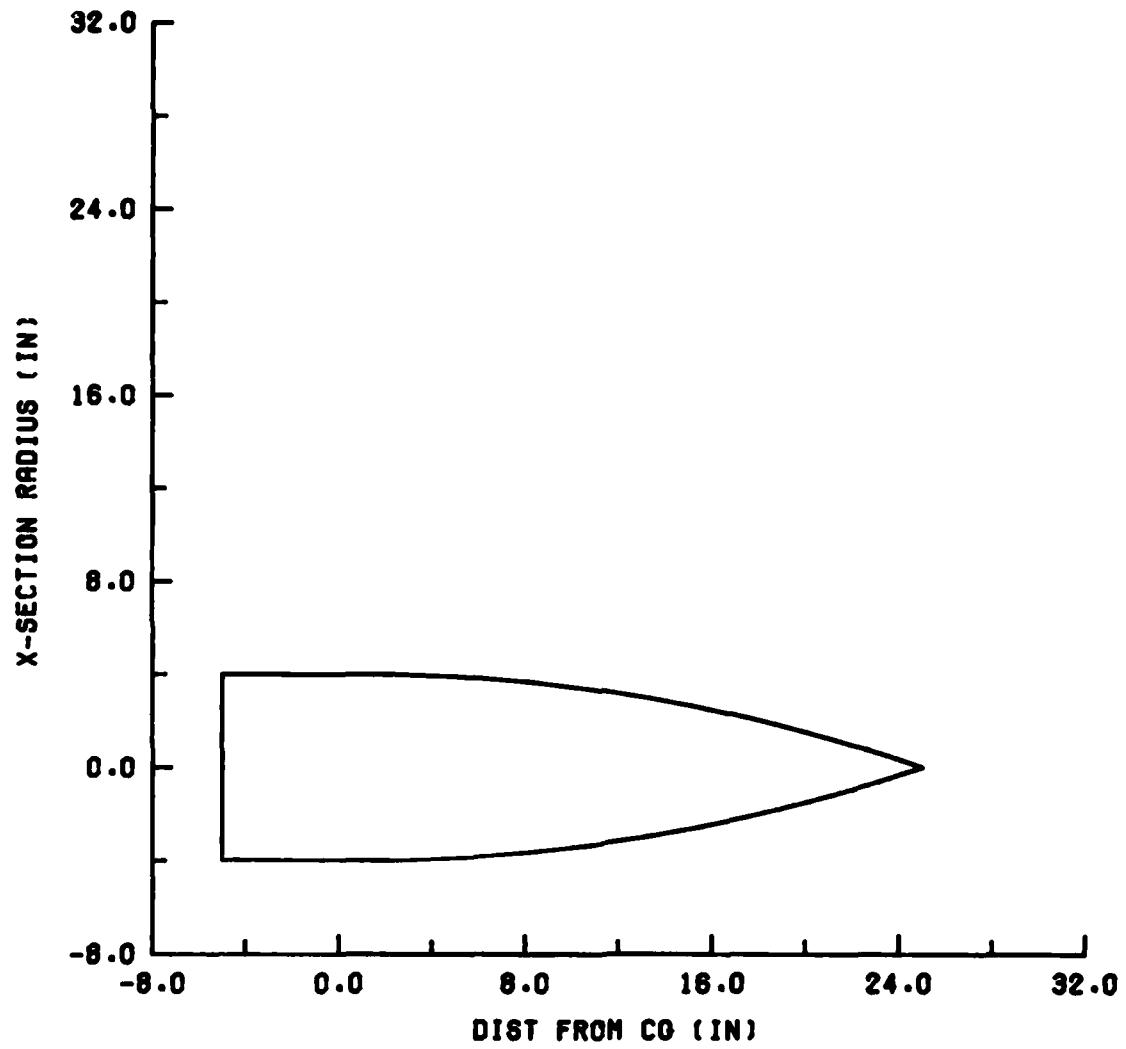


Figure 2.6 PENC02D-generated plots for sample problem 1 (Sheet 1 of 3).

SANDIA SLAB TEST R 339-106 (MAY 1970)
D=8.00 IN., W=670.00 LBS. V=695 FPS
ALPHA=0.00 DEG., GAMMA=180 DEG.

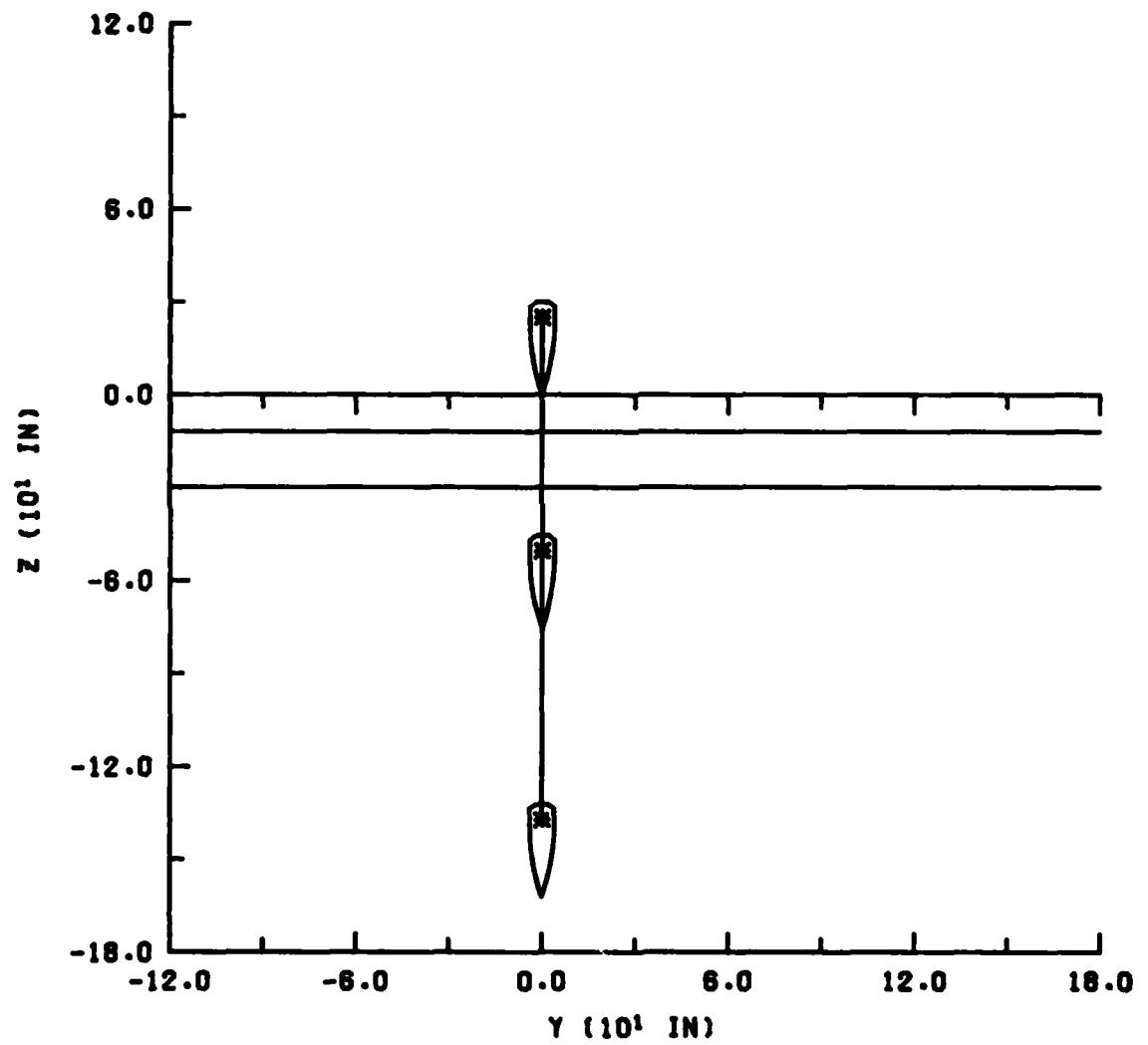


Figure 2.6 (Sheet 2 of 3).

SANDIA SLAB TEST R 339-106 (MAY 1970)
D=8.00 IN.. W=670.00 LBS. V=695 FPS
ALPHA=0.00 DEG.. GAMMA=180 DEG.

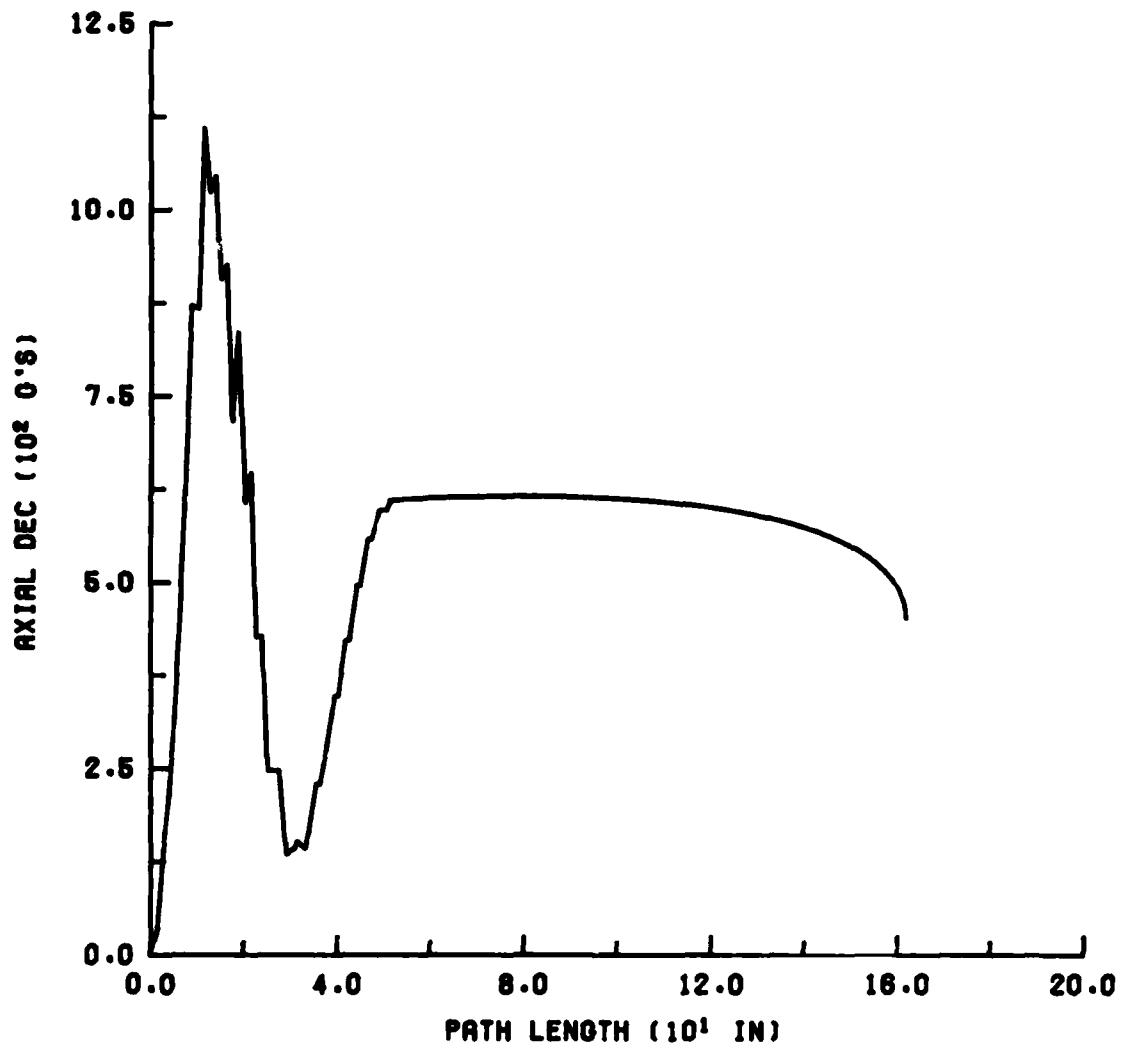


Figure 2.6 (Sheet 3 of 3).

BLUNTED CONE INTO S=5 SOIL (OBlique IMPACT)

```
*****
* MCS 2D NEWTONIAN PENETRATION CODE *
*****  

NEMOV2= 10  TOPSHR= 4  NSTP= 190  NLAY= 2  NMPLT3= 3  NMSTPZ= 3  NMUSC= 20  NMUDC= 0  NMINT= 2  

ADDITIONAL PLOTS... 2 3 0  

DELS1(1)= 0.  

RB= 0.100000E 01  EJ= 0.  ARATIO= 0.125000E 01  ELAS= 0.100000E-01  DGMAD= 0.150000E 02  

RJ= 0.  RN= 0.  XFF= 0.  

RC= 0.  RO= 0.300000E 01  XFA= 0.  

XLP= 0.200000E 02  XCG= 0.107070E 02  SN= 0.500000E 01  THETND= 0.216014E 02  

THETFD= 0.  VEL= 0.900000E 04  FREKT= 0.200000E 01  DLFAD= 0.  

GAMSTP= 0.360000E 03  YSTOP= 0.100000E 07  ZSTOP= 0.100000E 07  ZSHIFT= 0.  

FAD= 0.250000E 00  FANG= 0.100000E-01  DTIN= 0.100000E-07  DTMAX= 0.100000E 00  

WEIGHT= 0.132550E 03  XICG= 0.373300E 04  ALPHAD= 0.  

TIMEI= 0.  TIMEF= 0.100000E 02  FREQI= 0.200000E 01  GAMAD= 0.230000E 03  

PHIMIN= 0.300000E 01 DEG.  VELF= 0.200000E 03  WIX= 0.  GAMAD= 0.230000E 03  

*****
LAYER  (YOUNG'S S-NUMBER) (LB-DECSIT)N4)  {PSIP}  (LAYER DEPTH-IN.) (TENS. KIP/INDEX)  

1  * * * AIR * * * NO RESISTANCE * * *  

2      5.00      0.      0.      1.000E 06      10.0

```

Figure 2.7 Selected pages from the printed output for sample problem 2
(Sheet 1 of 9).

Figure 2.7 (Sheet 2 of 9).

```

STEP = 1
TIME = 0.
Y-POSITION = -0.869351E-01
Y-VELOCITY = -0.869445E-01
Y-FORCE = -0.132308E-02
Y-G'S = -0.188940E-05
AXIAL FORCE = 0.

Z-POSITION = -0.705131E-01
Z-VELOCITY = -0.705131E-01
Z-FORCE = -0.142541E-03
Z-G'S = -0.142541E-05
AXIAL G'S = 0.

PITCH ANGLE = 0.230000E-03
PITCH RATE = 0.620345E-03
PITCH FORCE = 0.1507E-05
PITCH MOMENT = -0.1507E-05
PROJ WEIGHT= 1.3255E-02

STEP = 2
TIME = 0.2502E-03
Y-POSITION = -0.667685E-01
Y-VELOCITY = -0.689095E-01
Y-FORCE = -0.126967E-02
Y-G'S = -0.175234E-05
AXIAL FORCE = 0.

Z-POSITION = -0.560810E-01
Z-VELOCITY = -0.574105E-01
Z-FORCE = -0.100793E-02
Z-G'S = -0.148323E-05
AXIAL G'S = 0.

PITCH ANGLE = 0.204950E-03
PITCH RATE = 0.605151E-03
PITCH FORCE = 0.156120E-05
PITCH MOMENT = -0.156120E-05
PROJ WEIGHT= 1.3255E-02

STEP = 3
TIME = 0.4776E-03
Y-POSITION = -0.514909E-01
Y-VELOCITY = -0.527070E-01
Y-FORCE = -0.940704E-02
Y-G'S = -0.129403E-05
AXIAL FORCE = 0.

Z-POSITION = -0.437300E-01
Z-VELOCITY = -0.457650E-01
Z-FORCE = -0.800000E-02
Z-G'S = -0.111441E-05
AXIAL G'S = 0.

PITCH ANGLE = 0.197495E-03
PITCH RATE = 0.596405E-03
PITCH FORCE = 0.146020E-05
PITCH MOMENT = -0.146020E-05
PROJ WEIGHT= 1.3255E-02

STEP = 4
TIME = 0.7543E-03
Y-POSITION = -0.371119E-01
Y-VELOCITY = -0.385310E-01
Y-FORCE = -0.716151E-02
Y-G'S = -0.109176E-05
AXIAL FORCE = 0.

Z-POSITION = -0.275165E-01
Z-VELOCITY = -0.289097E-01
Z-FORCE = -0.627526E-02
Z-G'S = -0.952678E-05
AXIAL G'S = 0.

PITCH ANGLE = 0.206441E-03
PITCH RATE = 0.594913E-03
PITCH FORCE = 0.147432E-05
PITCH MOMENT = -0.147432E-05
PROJ WEIGHT= 1.3255E-02

STEP = 5
TIME = 0.9239E-03
Y-POSITION = -0.295080E-01
Y-VELOCITY = -0.308937E-01
Y-FORCE = -0.594020E-02
Y-G'S = -0.105980E-06
AXIAL FORCE = 0.

Z-POSITION = -0.190781E-01
Z-VELOCITY = -0.193268E-01
Z-FORCE = -0.502895E-02
Z-G'S = -0.792954E-06
AXIAL G'S = 0.

PITCH ANGLE = 0.205929E-03
PITCH RATE = 0.595053E-03
PITCH FORCE = 0.148633E-05
PITCH MOMENT = -0.148633E-05
PROJ WEIGHT= 1.3255E-02

STEP = 6
TIME = 0.1100E-02
Y-POSITION = -0.399176E-01
Y-VELOCITY = -0.412681E-01
Y-FORCE = -0.713681E-02
Y-G'S = -0.111368E-06
AXIAL FORCE = 0.

Z-POSITION = -0.297494E-01
Z-VELOCITY = -0.310467E-01
Z-FORCE = -0.584202E-02
Z-G'S = -0.917366E-06
AXIAL G'S = 0.

PITCH ANGLE = 0.215012E-03
PITCH RATE = 0.595324E-03
PITCH FORCE = 0.149733E-05
PITCH MOMENT = -0.149733E-05
PROJ WEIGHT= 1.3255E-02

STEP = 7
TIME = 0.1438E-02
Y-POSITION = -0.473671E-01
Y-VELOCITY = -0.473749E-01
Y-FORCE = -0.800000E-02
Y-G'S = -0.121597E-06
AXIAL FORCE = 0.

Z-POSITION = -0.194777E-01
Z-VELOCITY = -0.194742E-01
Z-FORCE = -0.501636E-02
Z-G'S = -0.917366E-06
AXIAL G'S = 0.

PITCH ANGLE = 0.209163E-03
PITCH RATE = 0.594905E-03
PITCH FORCE = 0.149733E-05
PITCH MOMENT = -0.149733E-05
PROJ WEIGHT= 1.3255E-02

STEP = 8
TIME = 0.1612E-02
Y-POSITION = -0.544771E-01
Y-VELOCITY = -0.560105E-01
Y-FORCE = -0.842420E-02
Y-G'S = -0.125840E-06
AXIAL FORCE = 0.

Z-POSITION = -0.194777E-01
Z-VELOCITY = -0.194742E-01
Z-FORCE = -0.501636E-02
Z-G'S = -0.917366E-06
AXIAL G'S = 0.

PITCH ANGLE = 0.224693E-03
PITCH RATE = 0.594944E-03
PITCH FORCE = 0.149733E-05
PITCH MOMENT = -0.149733E-05
PROJ WEIGHT= 1.3255E-02

STEP = 9
TIME = 0.1875E-02
Y-POSITION = -0.614771E-01
Y-VELOCITY = -0.627338E-01
Y-FORCE = -0.877689E-02
Y-G'S = -0.120945E-06
AXIAL FORCE = 0.

Z-POSITION = -0.324963E-01
Z-VELOCITY = -0.313434E-01
Z-FORCE = -0.510134E-02
Z-G'S = -0.912444E-06
AXIAL G'S = 0.

PITCH ANGLE = 0.230744E-03
PITCH RATE = 0.594944E-03
PITCH FORCE = 0.149733E-05
PITCH MOMENT = -0.149733E-05
PROJ WEIGHT= 1.3255E-02

STEP = 10
TIME = 0.2141E-02
Y-POSITION = -0.683649E-01
Y-VELOCITY = -0.704652E-01
Y-FORCE = -0.911411E-02
Y-G'S = -0.1141E-06
AXIAL FORCE = 0.

Z-POSITION = -0.474066E-01
Z-VELOCITY = -0.459931E-01
Z-FORCE = -0.510745E-02
Z-G'S = -0.907453E-06
AXIAL G'S = 0.

PITCH ANGLE = 0.235016E-03
PITCH RATE = 0.594944E-03
PITCH FORCE = 0.149733E-05
PITCH MOMENT = -0.149733E-05
PROJ WEIGHT= 1.3255E-02

```

Figure 2.7 (Sheet 3 of 9).

Figure 2.7 (Sheet 4 of 9).

STEP = 1	Y-POSITION	0.9422E-01	PITCH ANGLE	0.245000E 03	COUNT	0
TIME = 0.	Y-FORCE	0.4000E-02	PITCH FORCE	0.466818E 05	ROLL ALPHA	0.02778E-04
	Z-POSITION	0.4000E-02	PITCH MOMENT	0.442372E 05	ROLL WEIGHT	1.3255E 02
	Z-VELOCITY	0.0000E 00	PITCH ANGLE	0.250000E 03	COUNT	1
	Z-LOCAL G'S	0.0000E 00	PITCH FORCE	0.440000E 05	ROLL ALPHA	0.02778E-04
	Z-LOCAL G'S	0.0000E 00	PITCH MOMENT	0.440000E 05	ROLL WEIGHT	1.3255E 02
STEP = 2	Y-POSITION	0.9017E-01	PITCH ANGLE	0.250000E 03	COUNT	2
TIME = 0.2503E-03	Y-FORCE	0.4000E-02	PITCH FORCE	0.440000E 05	ROLL ALPHA	0.02778E-04
	Z-POSITION	0.4000E-02	PITCH MOMENT	0.440000E 05	ROLL WEIGHT	1.3255E 02
	Z-VELOCITY	0.0000E 00	PITCH ANGLE	0.250000E 03	COUNT	3
	Z-LOCAL G'S	0.0000E 00	PITCH FORCE	0.440000E 05	ROLL ALPHA	0.02778E-04
	Z-LOCAL G'S	0.0000E 00	PITCH MOMENT	0.440000E 05	ROLL WEIGHT	1.3255E 02
STEP = 3	Y-POSITION	0.8974E-01	PITCH ANGLE	0.250000E 03	COUNT	4
TIME = 0.4739E-03	Y-FORCE	0.4000E-02	PITCH FORCE	0.440000E 05	ROLL ALPHA	0.02778E-04
	Z-POSITION	0.4000E-02	PITCH MOMENT	0.440000E 05	ROLL WEIGHT	1.3255E 02
	Z-VELOCITY	0.0000E 00	PITCH ANGLE	0.250000E 03	COUNT	5
	Z-LOCAL G'S	0.0000E 00	PITCH FORCE	0.440000E 05	ROLL ALPHA	0.02778E-04
	Z-LOCAL G'S	0.0000E 00	PITCH MOMENT	0.440000E 05	ROLL WEIGHT	1.3255E 02
STEP = 4	Y-POSITION	0.8932E-01	PITCH ANGLE	0.250000E 03	COUNT	6
TIME = 0.7550E-03	Y-FORCE	0.4000E-02	PITCH FORCE	0.440000E 05	ROLL ALPHA	0.02778E-04
	Z-POSITION	0.4000E-02	PITCH MOMENT	0.440000E 05	ROLL WEIGHT	1.3255E 02
	Z-VELOCITY	0.0000E 00	PITCH ANGLE	0.250000E 03	COUNT	7
	Z-LOCAL G'S	0.0000E 00	PITCH FORCE	0.440000E 05	ROLL ALPHA	0.02778E-04
	Z-LOCAL G'S	0.0000E 00	PITCH MOMENT	0.440000E 05	ROLL WEIGHT	1.3255E 02
STEP = 5	Y-POSITION	0.8890E-01	PITCH ANGLE	0.250000E 03	COUNT	8
TIME = 0.9245E-03	Y-FORCE	0.4000E-02	PITCH FORCE	0.440000E 05	ROLL ALPHA	0.02778E-04
	Z-POSITION	0.4000E-02	PITCH MOMENT	0.440000E 05	ROLL WEIGHT	1.3255E 02
	Z-VELOCITY	0.0000E 00	PITCH ANGLE	0.250000E 03	COUNT	9
	Z-LOCAL G'S	0.0000E 00	PITCH FORCE	0.440000E 05	ROLL ALPHA	0.02778E-04
	Z-LOCAL G'S	0.0000E 00	PITCH MOMENT	0.440000E 05	ROLL WEIGHT	1.3255E 02
STEP = 6	Y-POSITION	0.8848E-01	PITCH ANGLE	0.250000E 03	COUNT	10
TIME = 0.1160E-02	Y-FORCE	0.4000E-02	PITCH FORCE	0.440000E 05	ROLL ALPHA	0.02778E-04
	Z-POSITION	0.4000E-02	PITCH MOMENT	0.440000E 05	ROLL WEIGHT	1.3255E 02
	Z-VELOCITY	0.0000E 00	PITCH ANGLE	0.250000E 03	COUNT	11
	Z-LOCAL G'S	0.0000E 00	PITCH FORCE	0.440000E 05	ROLL ALPHA	0.02778E-04
	Z-LOCAL G'S	0.0000E 00	PITCH MOMENT	0.440000E 05	ROLL WEIGHT	1.3255E 02
STEP = 7	Y-POSITION	0.8806E-01	PITCH ANGLE	0.250000E 03	COUNT	12
TIME = 0.1438E-02	Y-FORCE	0.4000E-02	PITCH FORCE	0.440000E 05	ROLL ALPHA	0.02778E-04
	Z-POSITION	0.4000E-02	PITCH MOMENT	0.440000E 05	ROLL WEIGHT	1.3255E 02
	Z-VELOCITY	0.0000E 00	PITCH ANGLE	0.250000E 03	COUNT	13
	Z-LOCAL G'S	0.0000E 00	PITCH FORCE	0.440000E 05	ROLL ALPHA	0.02778E-04
	Z-LOCAL G'S	0.0000E 00	PITCH MOMENT	0.440000E 05	ROLL WEIGHT	1.3255E 02
STEP = 8	Y-POSITION	0.8764E-01	PITCH ANGLE	0.250000E 03	COUNT	14
TIME = 0.1611E-02	Y-FORCE	0.4000E-02	PITCH FORCE	0.440000E 05	ROLL ALPHA	0.02778E-04
	Z-POSITION	0.4000E-02	PITCH MOMENT	0.440000E 05	ROLL WEIGHT	1.3255E 02
	Z-VELOCITY	0.0000E 00	PITCH ANGLE	0.250000E 03	COUNT	15
	Z-LOCAL G'S	0.0000E 00	PITCH FORCE	0.440000E 05	ROLL ALPHA	0.02778E-04
	Z-LOCAL G'S	0.0000E 00	PITCH MOMENT	0.440000E 05	ROLL WEIGHT	1.3255E 02
STEP = 9	Y-POSITION	0.8722E-01	PITCH ANGLE	0.250000E 03	COUNT	16
TIME = 0.1673E-02	Y-FORCE	0.4000E-02	PITCH FORCE	0.440000E 05	ROLL ALPHA	0.02778E-04
	Z-POSITION	0.4000E-02	PITCH MOMENT	0.440000E 05	ROLL WEIGHT	1.3255E 02
	Z-VELOCITY	0.0000E 00	PITCH ANGLE	0.250000E 03	COUNT	17
	Z-LOCAL G'S	0.0000E 00	PITCH FORCE	0.440000E 05	ROLL ALPHA	0.02778E-04
	Z-LOCAL G'S	0.0000E 00	PITCH MOMENT	0.440000E 05	ROLL WEIGHT	1.3255E 02
STEP = 10	Y-POSITION	0.8680E-01	PITCH ANGLE	0.250000E 03	COUNT	18
TIME = 0.2137E-02	Y-FORCE	0.4000E-02	PITCH FORCE	0.440000E 05	ROLL ALPHA	0.02778E-04
	Z-POSITION	0.4000E-02	PITCH MOMENT	0.440000E 05	ROLL WEIGHT	1.3255E 02
	Z-VELOCITY	0.0000E 00	PITCH ANGLE	0.250000E 03	COUNT	19
	Z-LOCAL G'S	0.0000E 00	PITCH FORCE	0.440000E 05	ROLL ALPHA	0.02778E-04
	Z-LOCAL G'S	0.0000E 00	PITCH MOMENT	0.440000E 05	ROLL WEIGHT	1.3255E 02

Figure 2.7 (Sheet 5 of 9).

FINAL STATISTICS
 PATH LENGTH TRAVELED = 0.15537E 03 IN. (0.12990E 02 FT.)
 PATH LENGTH = 0.15537E 03 IN. (0.12990E 02 FT.)
 FINAL VELOCITY = 0.14614E 02 FT/SEC
 FINAL ACCELERATION = 0.95277E 01 G-S
 MAX. LATERAL ACCELERATION = 0.16136E 03 G-S
 MAX. LOCAL ACCELERATION = 0.87160E 03 G-S

Figure 2.7 (Sheet 6 of 9).

```

STEP = 1
TIME = 0.
Y-POSITION 0.2504E-03
Y-VELOCITY 0.2504E-03
Y-FORCE 0.2504E-03
AXIAL FORCE 0.2504E-03
STEP = 2
TIME = 0.2504E-03
Y-POSITION 0.4740E-03
Y-VELOCITY 0.4740E-03
Y-FORCE 0.4740E-03
AXIAL FORCE 0.4740E-03
STEP = 3
TIME = 0.7550E-03
Y-POSITION 0.9245E-03
Y-VELOCITY 0.9245E-03
Y-FORCE 0.9245E-03
AXIAL FORCE 0.9245E-03
STEP = 4
TIME = 0.1100E-02
Y-POSITION 0.1437E-02
Y-VELOCITY 0.1437E-02
Y-FORCE 0.1437E-02
AXIAL FORCE 0.1437E-02
STEP = 5
TIME = 0.1609E-02
Y-POSITION 0.2079E-02
Y-VELOCITY 0.2079E-02
Y-FORCE 0.2079E-02
AXIAL FORCE 0.2079E-02
STEP = 6
TIME = 0.2131E-02
Y-POSITION 0.2712E-02
Y-VELOCITY 0.2712E-02
Y-FORCE 0.2712E-02
AXIAL FORCE 0.2712E-02
STEP = 7
TIME = 0.2600E-02
Y-POSITION 0.3345E-02
Y-VELOCITY 0.3345E-02
Y-FORCE 0.3345E-02
AXIAL FORCE 0.3345E-02
STEP = 8
TIME = 0.3000E-02
Y-POSITION 0.4000E-02
Y-VELOCITY 0.4000E-02
Y-FORCE 0.4000E-02
AXIAL FORCE 0.4000E-02
STEP = 9
TIME = 0.3469E-02
Y-POSITION 0.4669E-02
Y-VELOCITY 0.4669E-02
Y-FORCE 0.4669E-02
AXIAL FORCE 0.4669E-02
STEP = 10
TIME = 0.4000E-02
Y-POSITION 0.5300E-02
Y-VELOCITY 0.5300E-02
Y-FORCE 0.5300E-02
AXIAL FORCE 0.5300E-02

```

Figure 2.7 (Sheet 7 of 9).

```

STEP = 31
TIME = 0.7350E-02
Y-POSITION = 0.0000E+00
Y-VELOCITY = 0.0000E+00
Y-FORCE = 0.0000E+00
AXIAL FORCE = 0.0000E+00
STEP = 32
TIME = 0.7551E-02
Y-POSITION = 0.0000E+00
Y-VELOCITY = 0.0000E+00
Y-FORCE = 0.0000E+00
AXIAL FORCE = 0.0000E+00
STEP = 33
TIME = 0.7642E-02
Y-POSITION = 0.0000E+00
Y-VELOCITY = 0.0000E+00
Y-FORCE = 0.0000E+00
AXIAL FORCE = 0.0000E+00
STEP = 34
TIME = 0.8133E-02
Y-POSITION = 0.0000E+00
Y-VELOCITY = 0.0000E+00
Y-FORCE = 0.0000E+00
AXIAL FORCE = 0.0000E+00
STEP = 35
TIME = 0.8327E-02
Y-POSITION = 0.0000E+00
Y-VELOCITY = 0.0000E+00
Y-FORCE = 0.0000E+00
AXIAL FORCE = 0.0000E+00
STEP = 36
TIME = 0.8616E-02
Y-POSITION = 0.0000E+00
Y-VELOCITY = 0.0000E+00
Y-FORCE = 0.0000E+00
AXIAL FORCE = 0.0000E+00
STEP = 37
TIME = 0.8909E-02
Y-POSITION = 0.0000E+00
Y-VELOCITY = 0.0000E+00
Y-FORCE = 0.0000E+00
AXIAL FORCE = 0.0000E+00
STEP = 38
TIME = 0.9103E-02
Y-POSITION = 0.0000E+00
Y-VELOCITY = 0.0000E+00
Y-FORCE = 0.0000E+00
AXIAL FORCE = 0.0000E+00
STEP = 39
TIME = 0.9394E-02
Y-POSITION = 0.0000E+00
Y-VELOCITY = 0.0000E+00
Y-FORCE = 0.0000E+00
AXIAL FORCE = 0.0000E+00
STEP = 40
TIME = 0.9685E-02
Y-POSITION = 0.0000E+00
Y-VELOCITY = 0.0000E+00
Y-FORCE = 0.0000E+00
AXIAL FORCE = 0.0000E+00

```

Figure 2.7 (Sheet 8 of 9).

```

STEP = 41
TIME = 0.9879E-02
Y-POSITION 0.77332E-02
Y-VELOCITY 0.0000E+00
AXIAL FORCE 0.0000E+00
STEP = 42
TIME = 0.1017E-01
Y-POSITION 0.72332E-02
Y-VELOCITY 0.0000E+00
AXIAL FORCE 0.0000E+00
STEP = 43
TIME = 0.1046E-01
Y-POSITION 0.72332E-02
Y-VELOCITY 0.0000E+00
AXIAL FORCE 0.0000E+00
STEP = 44
TIME = 0.1065E-01
Y-POSITION 0.72332E-02
Y-VELOCITY 0.0000E+00
AXIAL FORCE 0.0000E+00
STEP = 45
TIME = 0.1095E-01
Y-POSITION 0.72332E-02
Y-VELOCITY 0.0000E+00
AXIAL FORCE 0.0000E+00
STEP = 46
TIME = 0.1124E-01
Y-POSITION 0.72332E-02
Y-VELOCITY 0.0000E+00
AXIAL FORCE 0.0000E+00

```

```

FINAL STATISTICS
PATH LENGTH: 1.00628E-02 IN. ( 0.75523E 01 FT. )
TIME: 0.1124E-01 SEC
FINAL SPEED: 0.17337E 04 G'S
FINAL ACCELERATION: -0.17337E 04 G'S
MAX. AXIAL DECELERATION: 0.59075E 03 G'S

```

Figure 2.7 (Sheet 9 of 9).

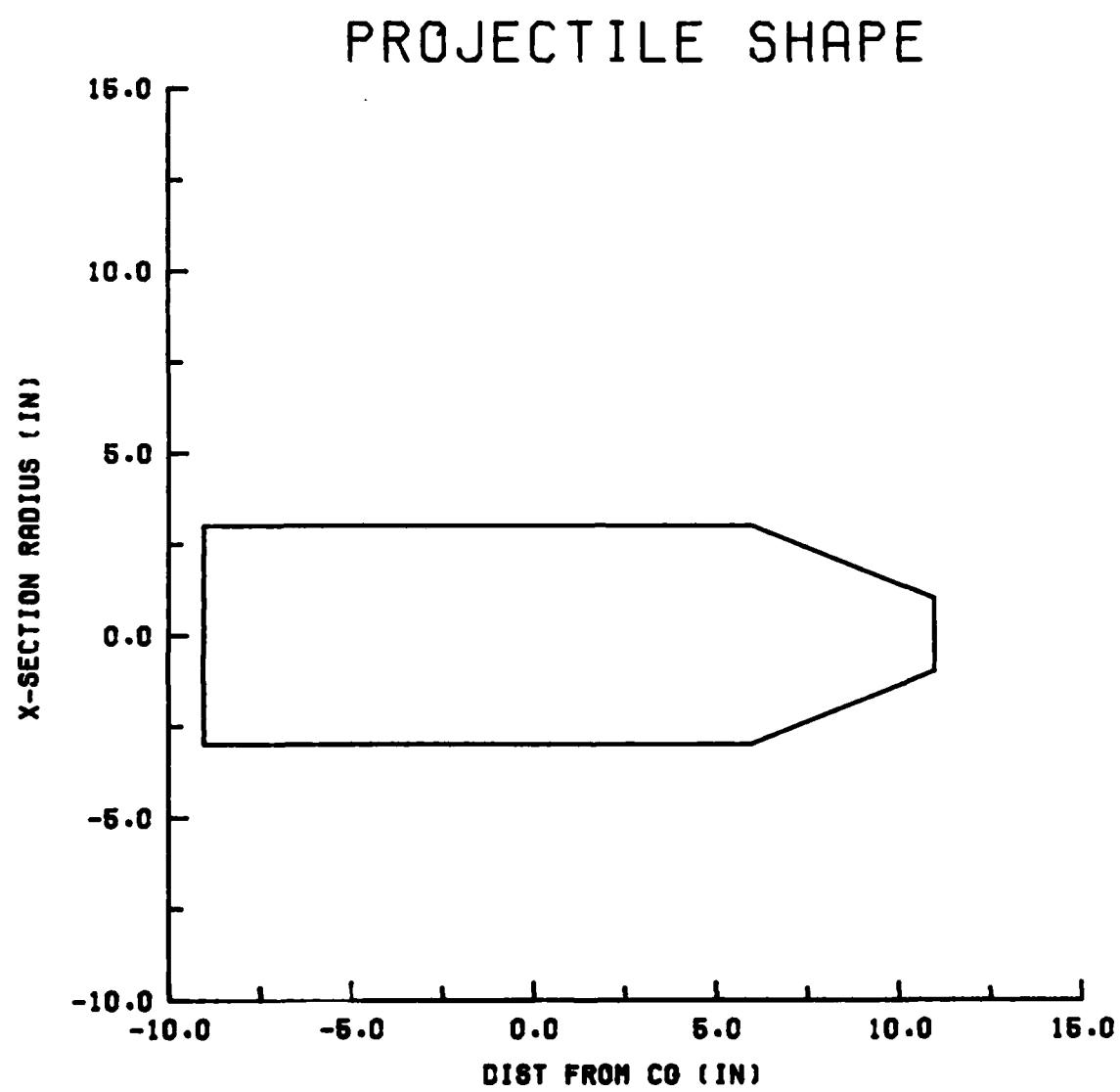


Figure 2.8 PENCO2D-generated plots for sample problem 2
(Sheet 1 of 13).

BLUNTED CONE INTO S=5 SOIL (OBLIQUE IMPACT)
D=6.00 IN., W=132.55 LBS. V=750 FPS
ALPHA=0.00 DEG., GAMMA=230 DEG.

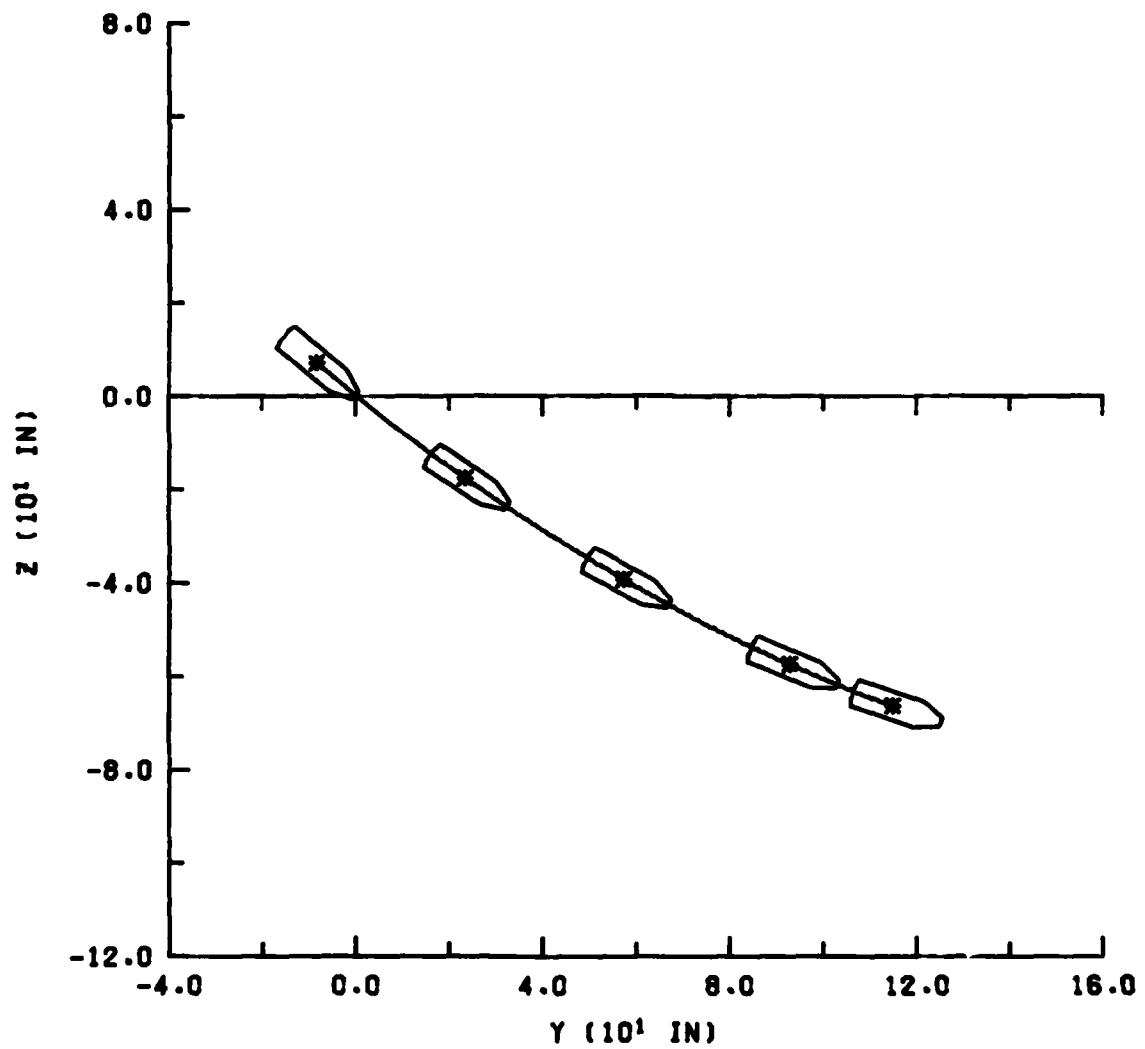


Figure 2.8 (Sheet 2 of 13).

BLUNTED CONE INTO S=5 SOIL (OBLIQUE IMPACT)
D=6.00 IN.. W=132.55 LBS. V=750 FPS
ALPHA=0.00 DEG., GAMMA=230 DEG.

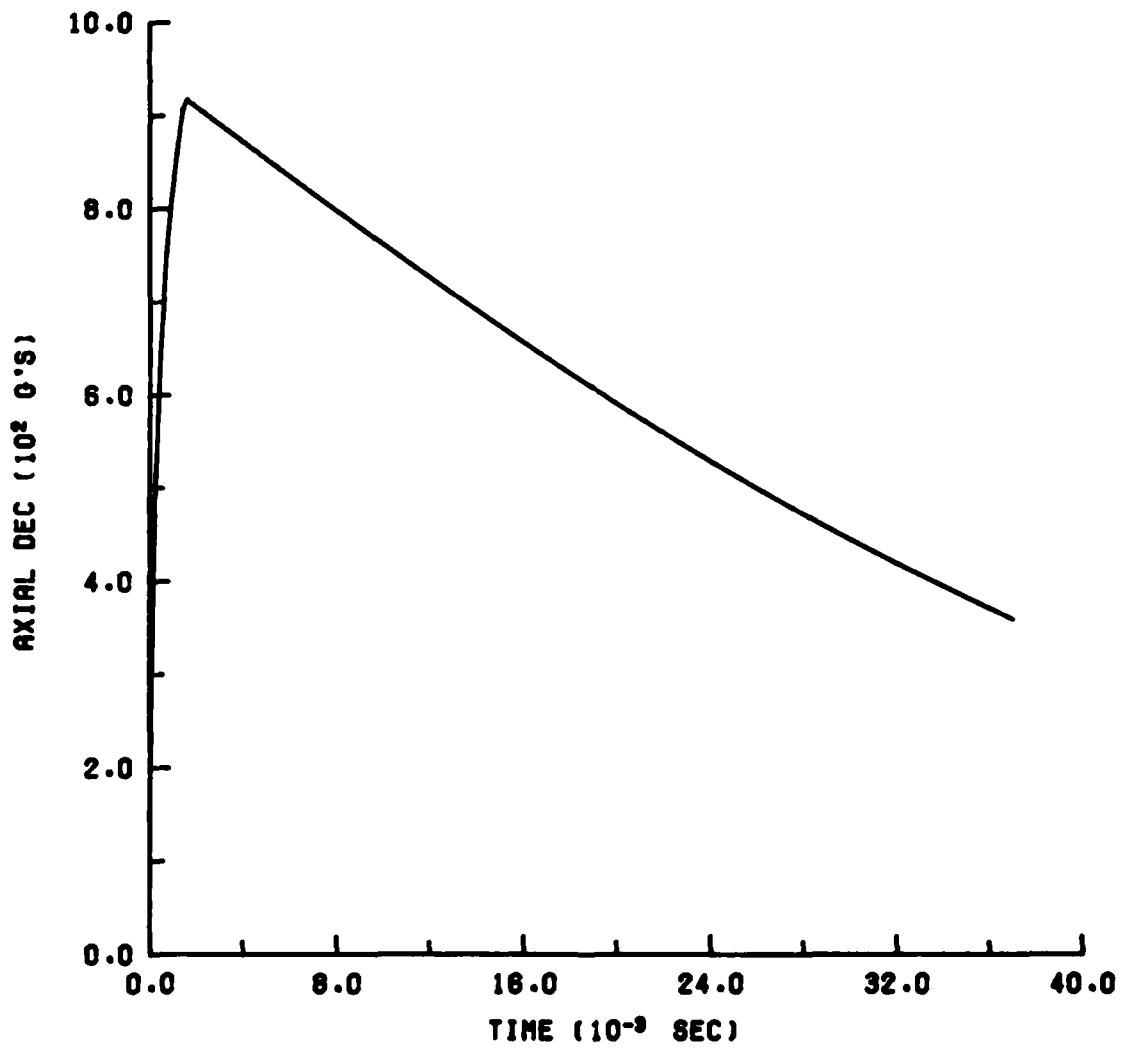


Figure 2.8 (Sheet 3 of 13).

BLUNTED CONE INTO S=5 SOIL (OBLIQUE IMPACT)
D=6.00 IN.. W=132.55 LBS. V=750 FPS
ALPHA=0.00 DEG.. GAMMA=230 DEG.

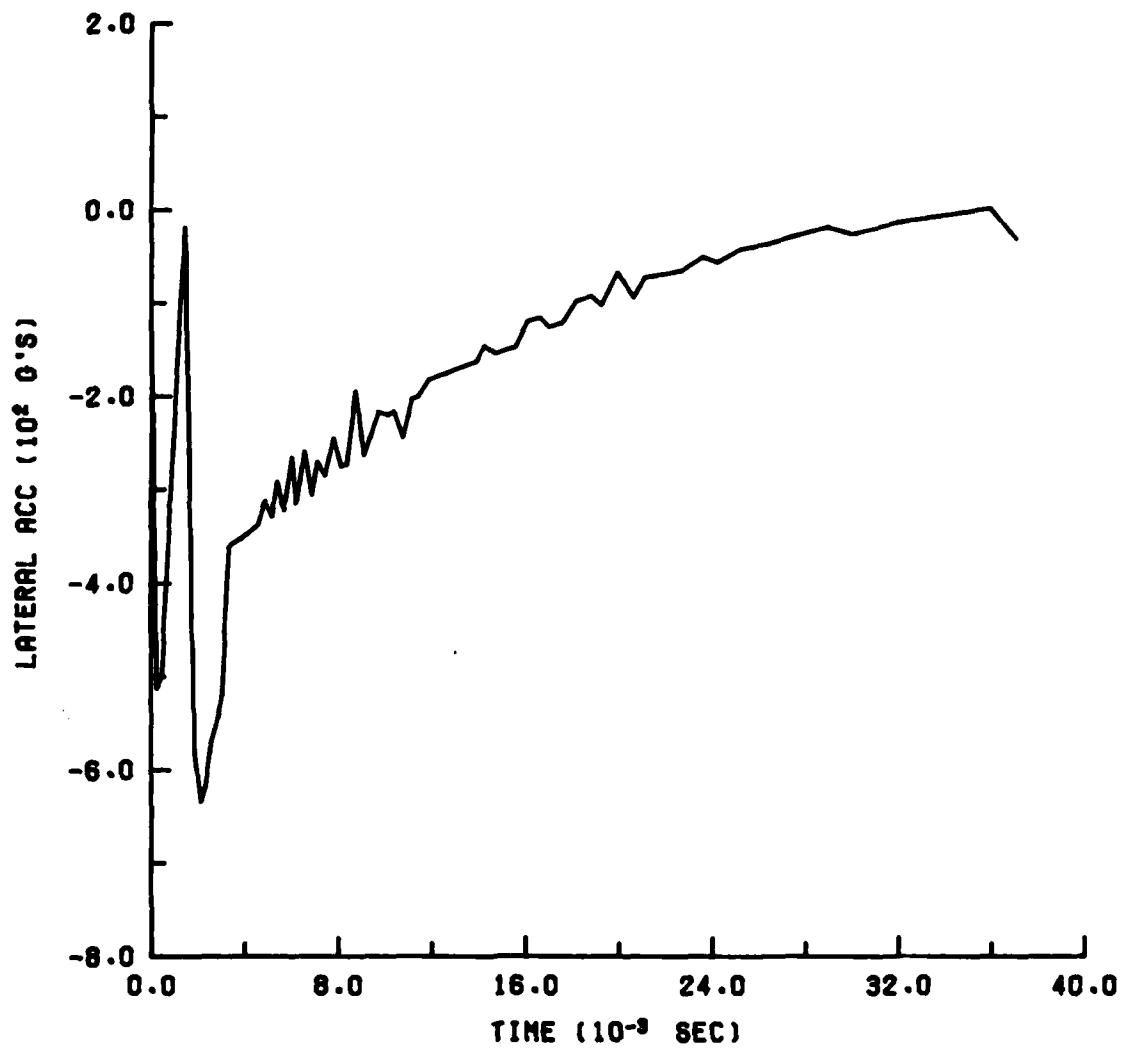


Figure 2.8 (Sheet 4 of 13).

BLUNTED CONE INTO S=5 SOIL (OBLIQUE IMPACT)
D=6.00 IN., W=132.55 LBS., V=750 FPS
ALPHA=0.00 DEG., GAMMA=230 DEG.

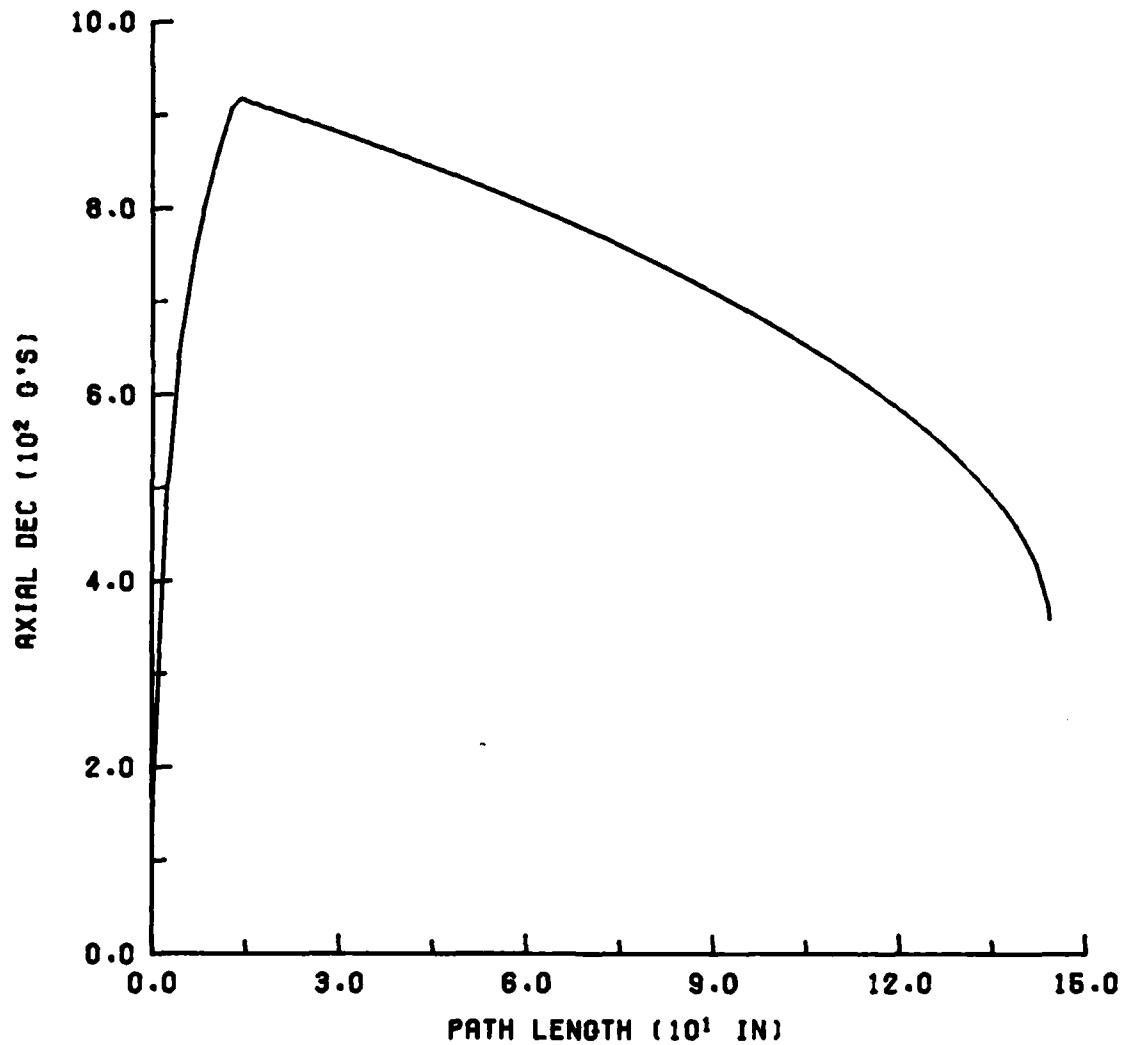


Figure 2.8 (Sheet 5 of 13).

BLUNTED CONE INTO S=5 SOIL (OBLIQUE IMPACT)
D=6.00 IN., W=132.55 LBS., V=750 FPS
ALPHA=0.00 DEG., GAMMA=245 DEG.

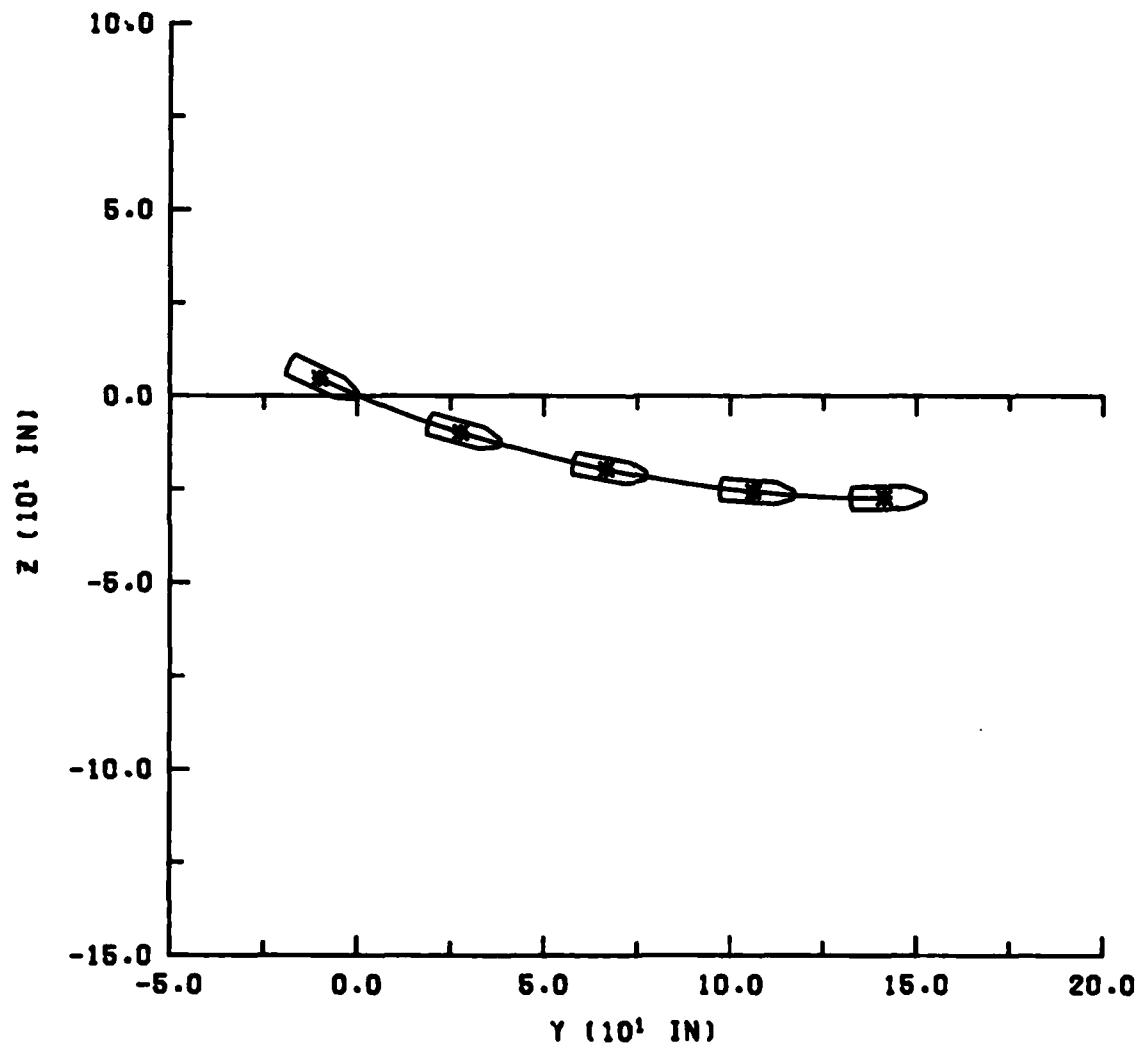


Figure 2.8 (Sheet 6 of 13).

BLUNTED CONE INTO S=5 SOIL (OBLIQUE IMPACT)
D=6.00 IN., W=132.55 LBS., V=750 FPS
ALPHA=0.00 DEG., GAMMA=245 DEG.

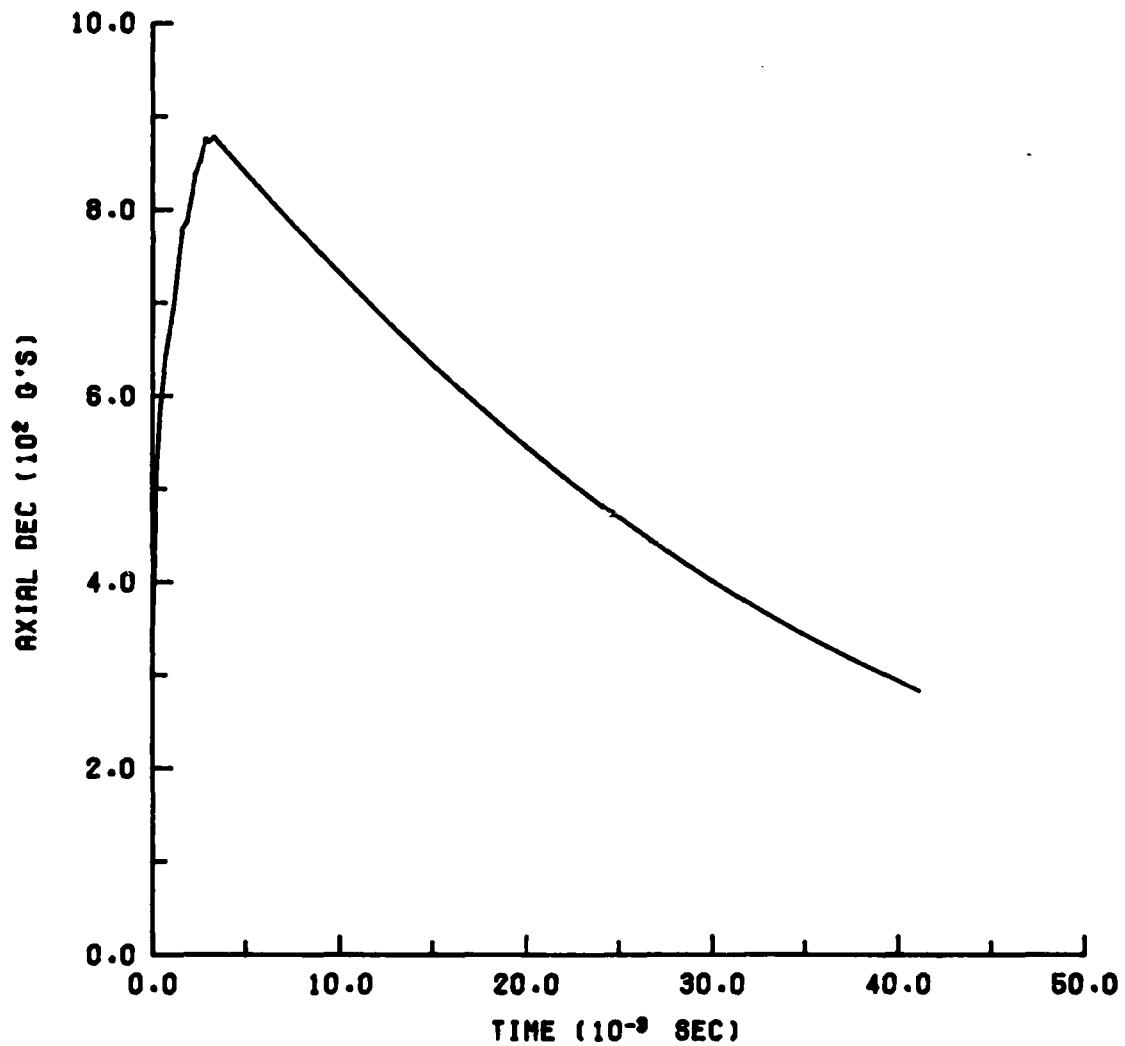


Figure 2.8 (Sheet 7 of 13).

BLUNTED CONE INTO S=5 SOIL (OBLIQUE IMPACT)
D=6.00 IN.. W=132.55 LBS. V=750 FPS
ALPHA=0.00 DEG., GAMMA=245 DEG.

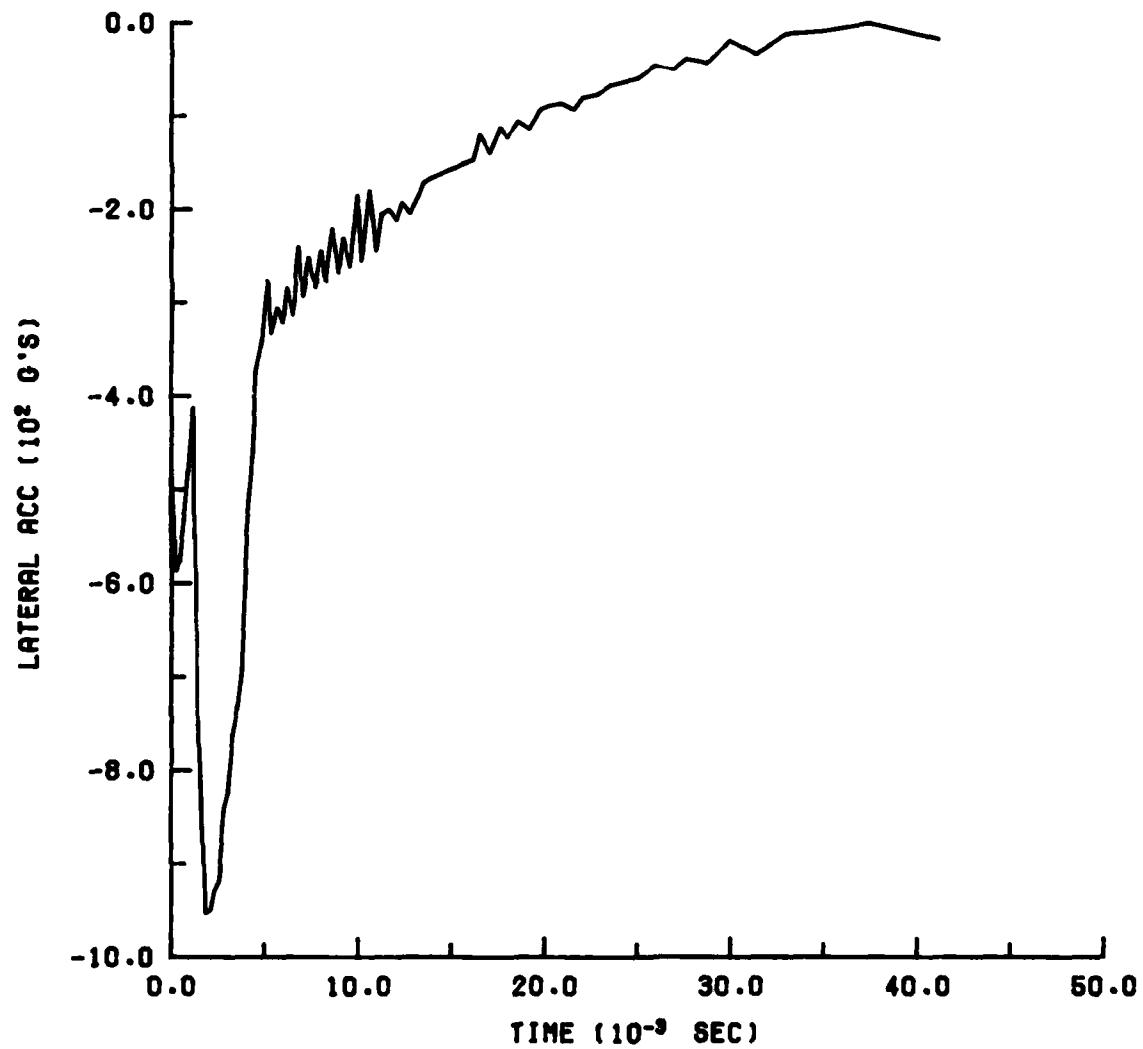


Figure 2.8 (Sheet 8 of 13).

BLUNTED CONE INTO S=5 SOIL (OBLIQUE IMPACT)
D=6.00 IN., W=132.55 LBS., V=750 FPS
ALPHA=0.00 DEG., GAMMA=245 DEG.

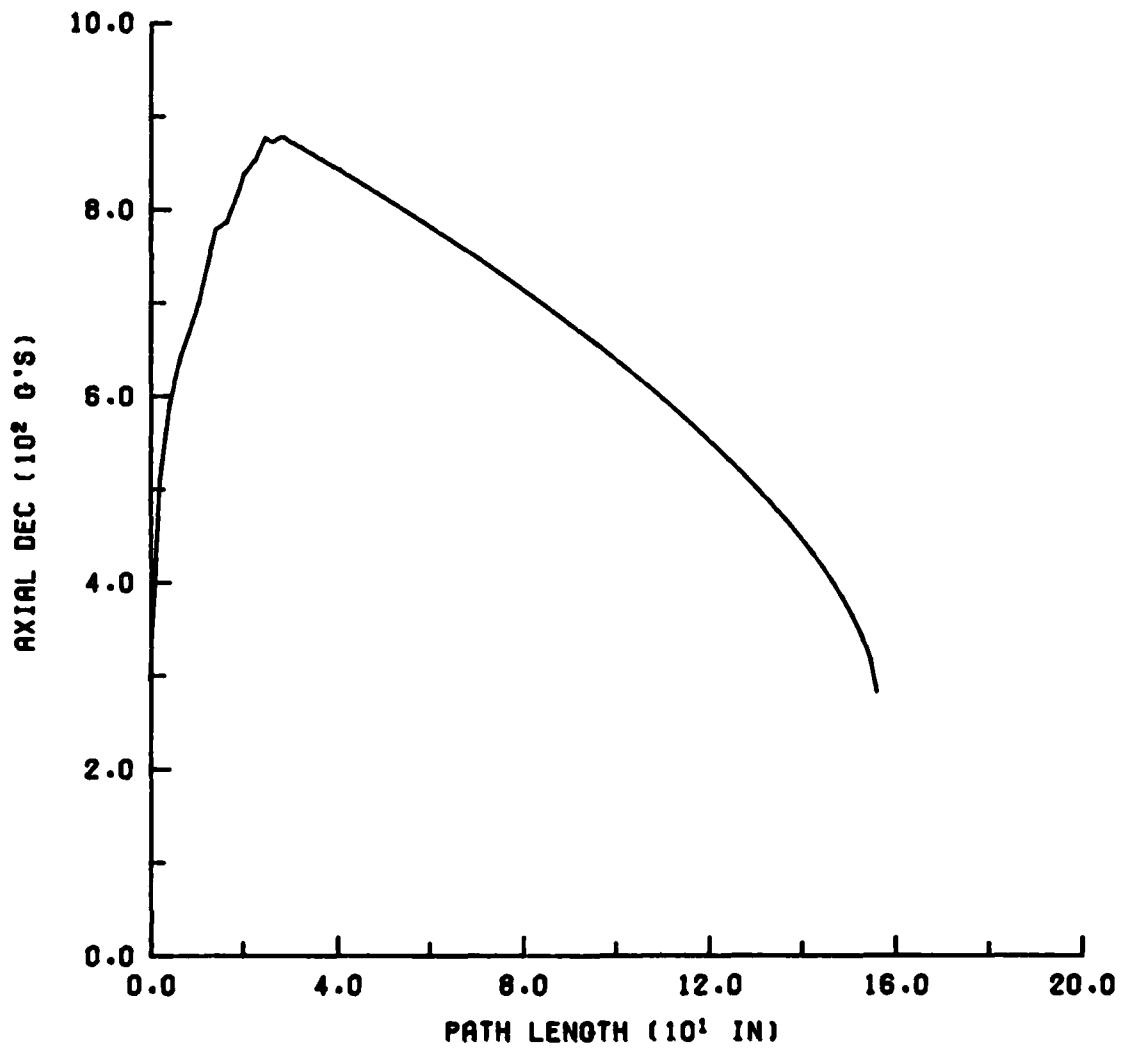


Figure 2.8 (Sheet 9 of 13).

BLUNTED CONE INTO S=5 SOIL (OBLIQUE IMPACT)
D=6.00 IN.. W=132.55 LBS. V=750 FPS
ALPHA=0.00 DEG., GAMMA=260 DEG.

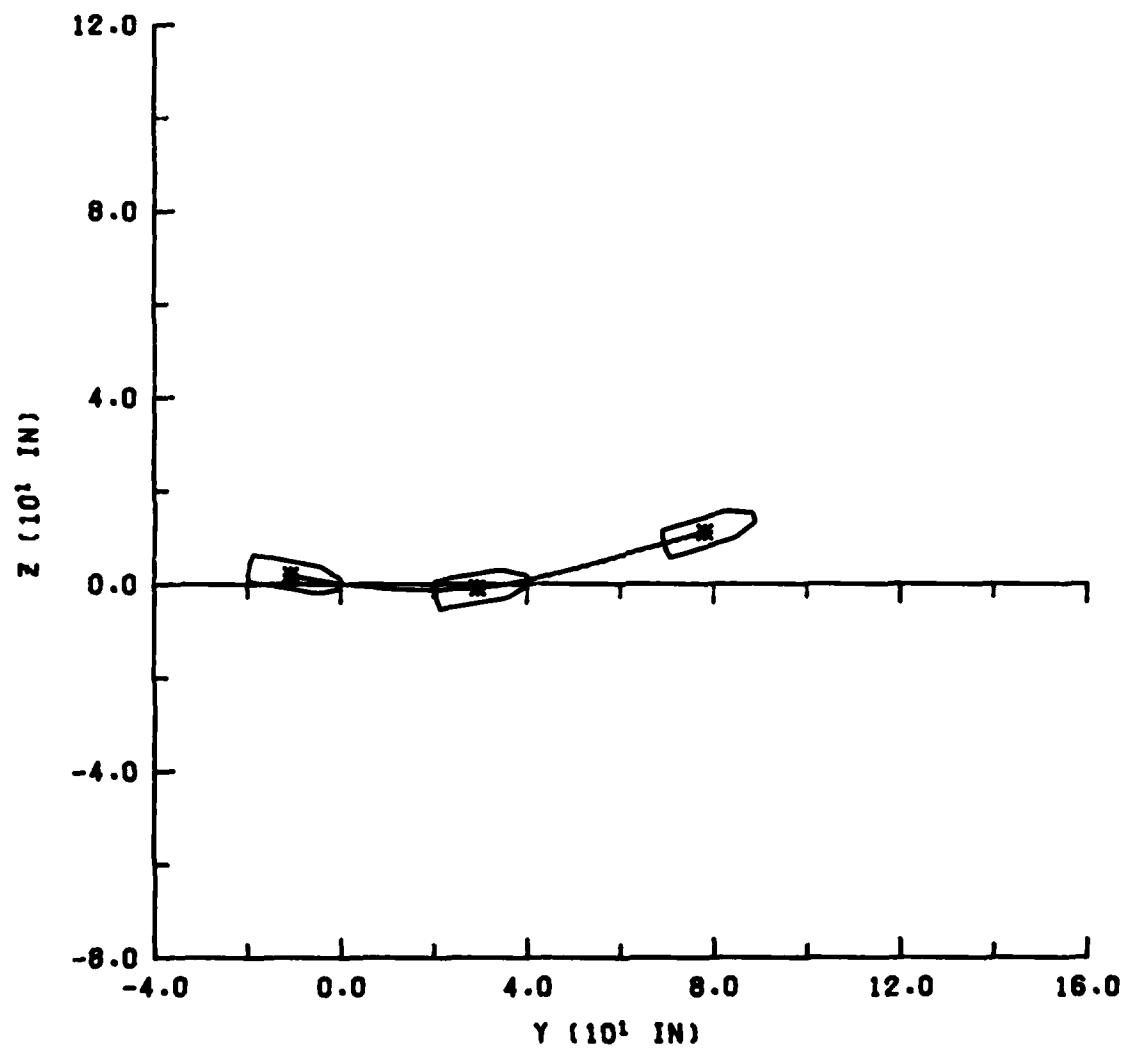


Figure 2.8 (Sheet 10 of 13).

BLUNTED CONE INTO S=5 SOIL (OBLIQUE IMPACT)
D=6.00 IN.. W=132.55 LBS. V=750 FPS
ALPHA=0.00 DEG.. GAMMA=260 DEG.

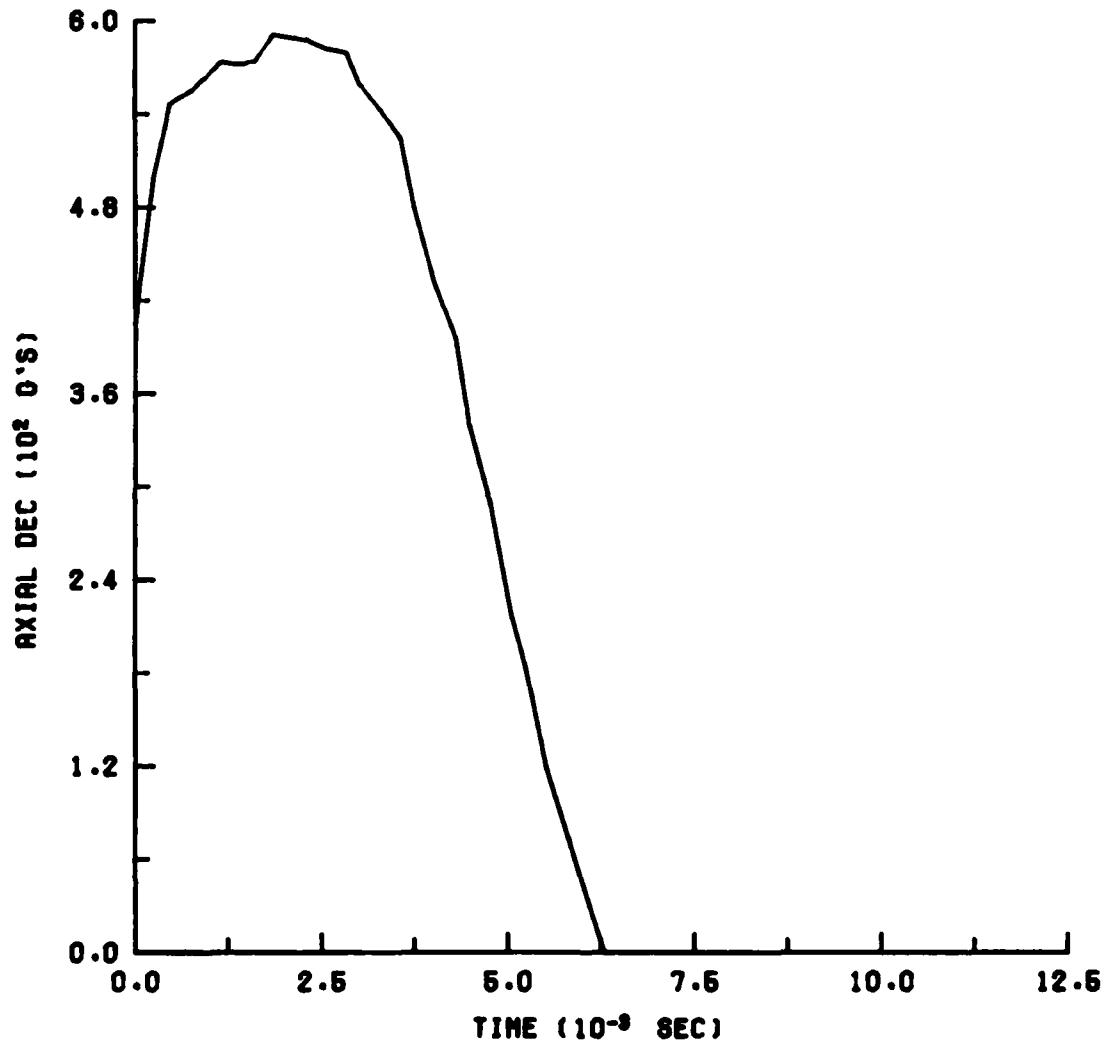


Figure 2.8 (Sheet 11 of 13).

BLUNTED CONE INTO S=5 SOIL (OBLIQUE IMPACT)
D=6.00 IN.. W=132.55 LBS. V=750 FPS
ALPHA=0.00 DEG.. GAMMA=260 DEG.

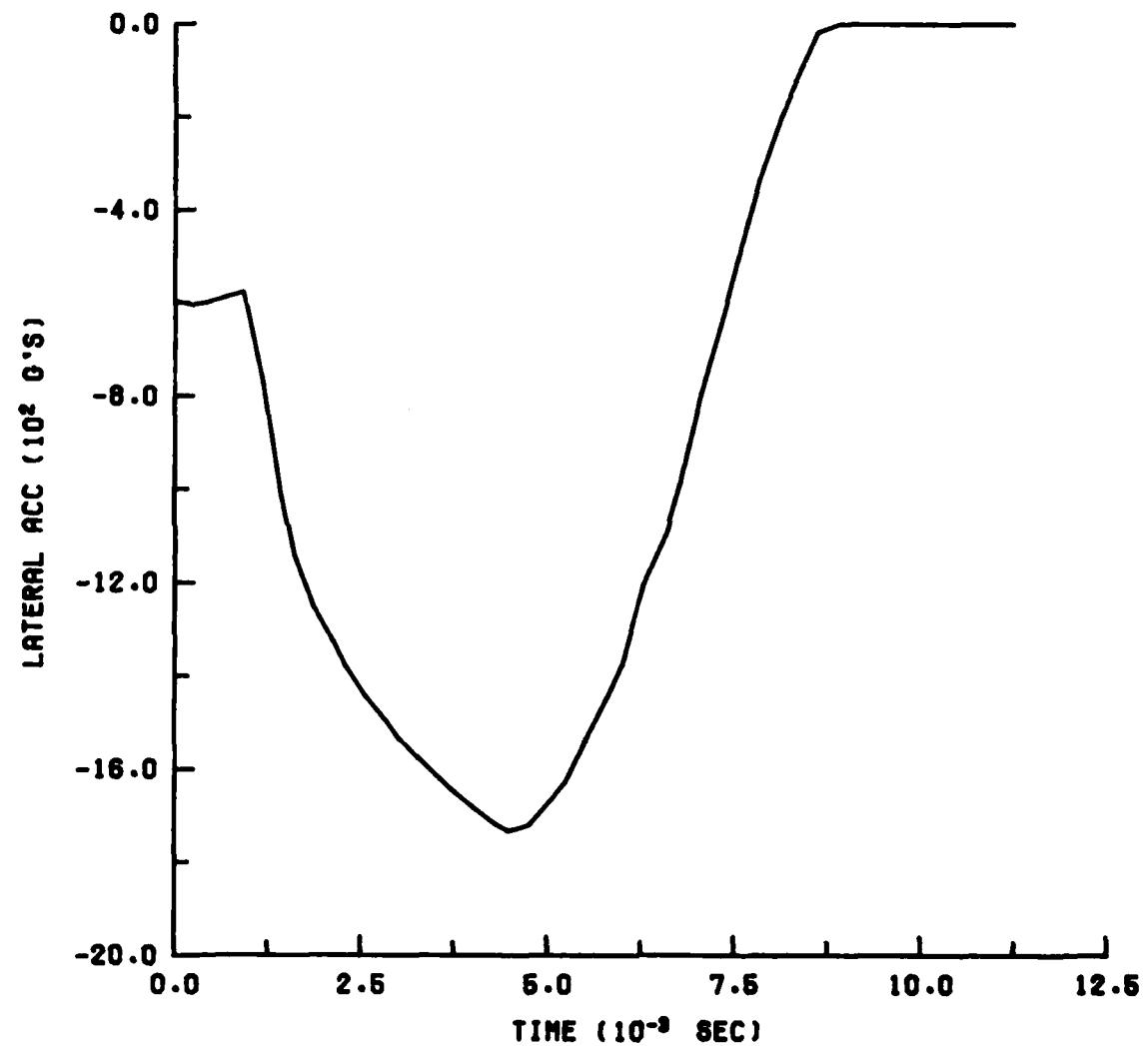


Figure 2.8 (Sheet 12 of 13).

BLUNTED CONE INTO S=5 SOIL (OBLIQUE IMPACT)
D=6.00 IN., W=132.55 LBS, V=750 FPS
ALPHA=0.00 DEG., GAMMA=260 DEG.

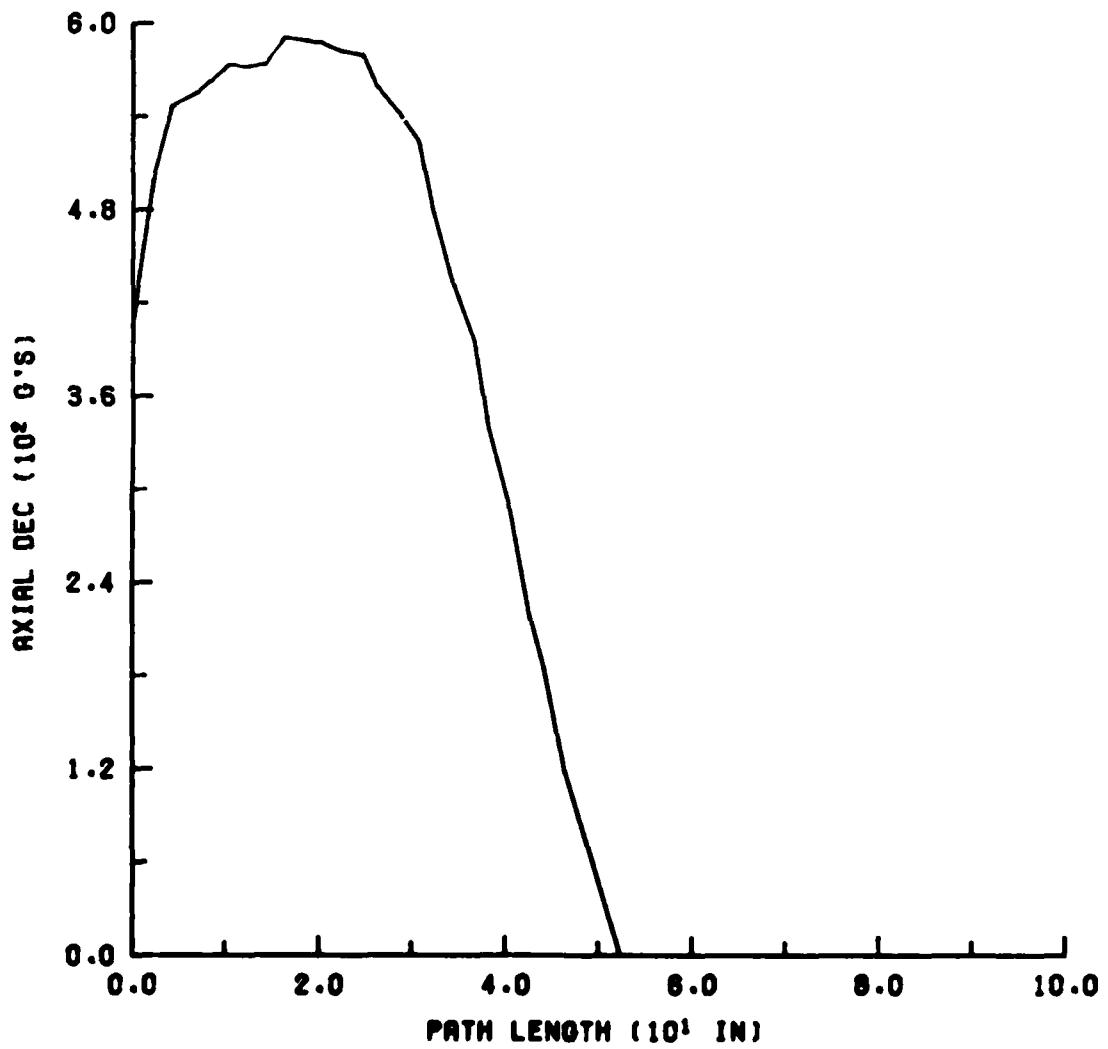
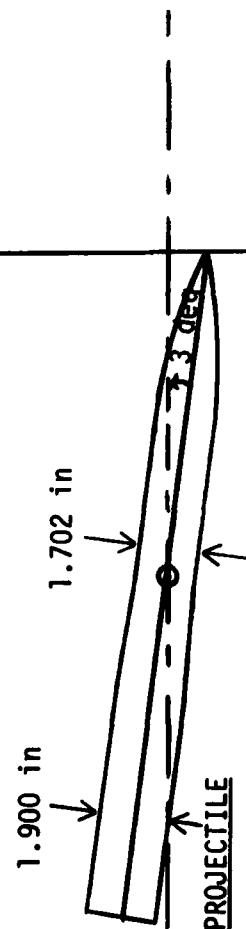
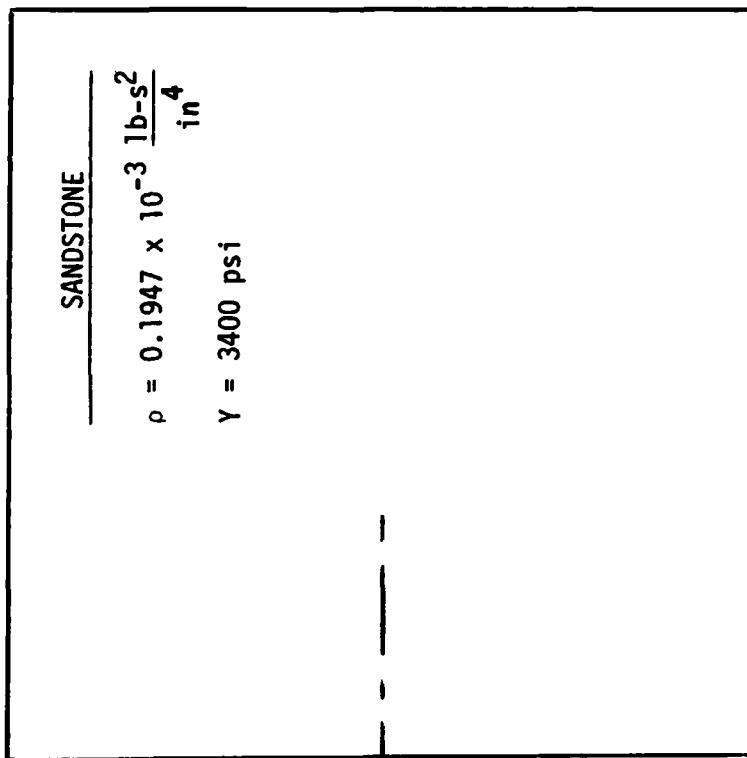


Figure 2.8 (Sheet 13 of 13).



Body:

$XCG = 8.71 \text{ in}$
 $XLP = 18.15 \text{ in}$
 $WEIGHT = 9.48 \text{ lb}$
 $XICG = 221.9 \text{ lb-in}^2$

Nose: Ogive with conical tip

$R0 = .851 \text{ in}$
 $RJ = 10.212 \text{ in}$
 $RB = .176 \text{ in}$
 $SN = 3.797 \text{ in}$
 $EJ = 0$

Figure 2.9 Nominal impact conditions for 3-degree RBT into sandstone.

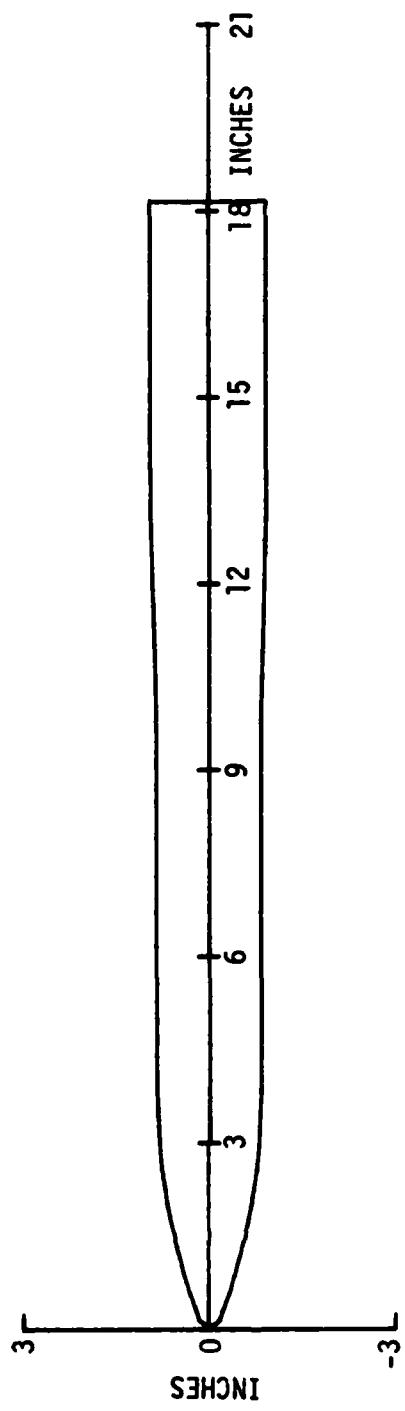


Figure 2.10 Surface geometry of actual projectile used in RBT's into sandstone.

```

SANDSTONE RBT SLED TEST...ALPHA=3 DEG

*-----*-----*-----*-----*-----*-----*-----*-----*-----*-----*
*   NEM 2D NEMO10.1AN PENETRATION CODE
*-----*-----*-----*-----*-----*-----*-----*-----*-----*-----*
NEM0V2= 6 10PSHP= 3 NSTPZ= 195 NLAY= 2 NMPLT3= 3 NMSTP= 1 NMUS= 150 TREDUC= 0 NPRINT= 1
DELS(1)= 0.250000E-01 ARATIO= 0.200000E 01 ELAS= 0.200000E-01 DGAMAD= 0.
R0= 0.176000E 00 EJ= 0. RJ= 0.102120E 02 RN= 0.
RC= 0. RG= 0.851000E 00 XFA= 0. XFF= 0.
XLP= 0.161500E 02 XCG= 0.971000E 01 SN= 0.379700E 01 THETAD= 0.
THETFD= 0. VEL= 0.160000E 05 FREKOT= 0.100000E-04 DALFAD= 0.
GAMSTP= 0.100000E 07 YSTOP= 0.100000E 07 ZSTOP=-0.100000E 07 ZSHIFT= 0.
FRA0= 0.200000E-01 FANG= 0.300000E-03 DTMAX= 0.200000E-07 DTMAX= 0.200000E-04
WEIGHT= 0.946000E 01 XICG= 0.221900E 03 ALPHAD= 0.300000E 01 ALSTP= 0.500000E 02
TIMEF= 0. 121000E-02 FREQI= 0.200000E 01
PMIN= 0. DEG. VELF= 0.240000E 03 MII= 0.
GAMAD= 0.162000E 03

LAYER NO. (YOUNG'S S-NUMBER) (LB-DEPTH/INCH) (YIELD) (LAYER DEPTH-IN.) (TENS.RIG.INDEX)
1 *** AIR *** NO RESISTANCE *** *
2 0. 1.9470E-04 3.400E 03 1.000E 06 350.0

```

Figure 2.11 Selected pages of the printed output for sample problem 3
(Sheet 1 of 4).

Figure 2.11 (Sheet 2 of 4).

```

STEP = 1
TIME = 0.
Y-POSITION = -0.455846E-03
Y-VELOCITY = 0.
Y-FORCE = 0.
Y-G'S = 0.
AXIAL FORCE = 0.

STEP = 2
TIME = 0.1000E-04
Y-POSITION = -0.455846E-03
Y-VELOCITY = 0.
Y-FORCE = 0.
AXIAL G'S = 0.

STEP = 3
TIME = 0.2000E-04
Y-POSITION = -0.455846E-03
Y-VELOCITY = 0.
Y-FORCE = 0.
AXIAL FORCE = 0.

STEP = 4
TIME = 0.3000E-04
Y-POSITION = -0.455846E-03
Y-VELOCITY = 0.
Y-FORCE = 0.
AXIAL G'S = 0.

STEP = 5
TIME = 0.4000E-04
Y-POSITION = -0.455846E-03
Y-VELOCITY = 0.
Y-FORCE = 0.
AXIAL FORCE = 0.

STEP = 6
TIME = 0.5000E-04
Y-POSITION = -0.455846E-03
Y-VELOCITY = 0.
Y-FORCE = 0.
AXIAL G'S = 0.

STEP = 7
TIME = 0.6000E-04
Y-POSITION = -0.455846E-03
Y-VELOCITY = 0.
Y-FORCE = 0.
AXIAL FORCE = 0.

STEP = 8
TIME = 0.7000E-04
Y-POSITION = -0.455846E-03
Y-VELOCITY = 0.
Y-FORCE = 0.
AXIAL G'S = 0.

STEP = 9
TIME = 0.8000E-04
Y-POSITION = -0.455846E-03
Y-VELOCITY = 0.
Y-FORCE = 0.
AXIAL FORCE = 0.

STEP = 10
TIME = 0.9000E-04
Y-POSITION = -0.455846E-03
Y-VELOCITY = 0.
Y-FORCE = 0.
AXIAL G'S = 0.

```

Figure 2.11 (Sheet 3 of 4).

```

STEP = 115          COUNT = 9          RETURN = 0.17260E-06
Y-POSITION = 0.11400E-02          DELTA = 0.3729E-06
Y-VELOCITY = 0.11400E-02          DIA = 0.32460E-06
AXIAL FORCE = 0.00000E+00          PROJ WEIGHT = 0.4800E-06

STEP = 116          COUNT = 9          RETURN = 0.17260E-06
Y-POSITION = 0.11500E-02          DELTA = 0.3729E-06
Y-VELOCITY = 0.11500E-02          DIA = 0.32460E-06
AXIAL FORCE = 0.00000E+00          PROJ WEIGHT = 0.4800E-06

STEP = 117          COUNT = 9          RETURN = 0.17260E-06
Y-POSITION = 0.11600E-02          DELTA = 0.3729E-06
Y-VELOCITY = 0.11600E-02          DIA = 0.32460E-06
AXIAL FORCE = 0.00000E+00          PROJ WEIGHT = 0.4800E-06

STEP = 118          COUNT = 9          RETURN = 0.17260E-06
Y-POSITION = 0.11700E-02          DELTA = 0.3729E-06
Y-VELOCITY = 0.11700E-02          DIA = 0.32460E-06
AXIAL FORCE = 0.00000E+00          PROJ WEIGHT = 0.4800E-06

STEP = 119          COUNT = 9          RETURN = 0.17260E-06
Y-POSITION = 0.11800E-02          DELTA = 0.3729E-06
Y-VELOCITY = 0.11800E-02          DIA = 0.32460E-06
AXIAL FORCE = 0.00000E+00          PROJ WEIGHT = 0.4800E-06

STEP = 120          COUNT = 9          RETURN = 0.17260E-06
Y-POSITION = 0.11900E-02          DELTA = 0.3729E-06
Y-VELOCITY = 0.11900E-02          DIA = 0.32460E-06
AXIAL FORCE = 0.00000E+00          PROJ WEIGHT = 0.4800E-06

STEP = 121          COUNT = 9          RETURN = 0.17260E-06
Y-POSITION = 0.12000E-02          DELTA = 0.3729E-06
Y-VELOCITY = 0.12000E-02          DIA = 0.32460E-06
AXIAL FORCE = 0.00000E+00          PROJ WEIGHT = 0.4800E-06

STEP = 122          COUNT = 9          RETURN = 0.17260E-06
Y-POSITION = 0.12100E-02          DELTA = 0.3729E-06
Y-VELOCITY = 0.12100E-02          DIA = 0.32460E-06
AXIAL FORCE = 0.00000E+00          PROJ WEIGHT = 0.4800E-06

```

Figure 2.11 (Sheet 4 of 4).

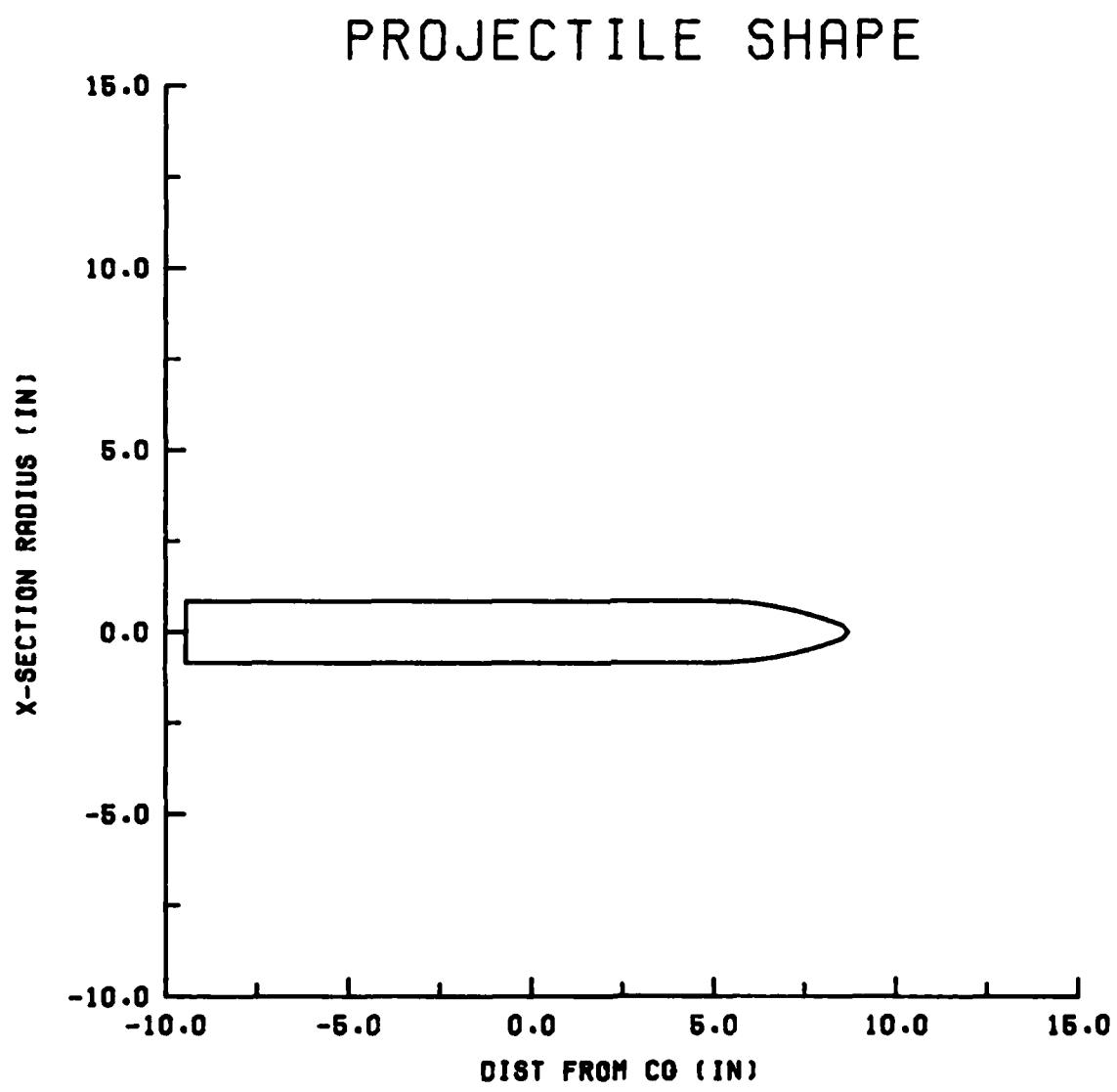


Figure 2.12 PENC02D-generated plots for sample problem 3
(Sheet 1 of 5).

SANDSTONE RBT SLED TEST...ALPHA=3 DEG
D=1.70 IN.. W=9.48 LBS. V=1500 FPS
ALPHA=3.00 DEG., GAMMA=183 DEG.

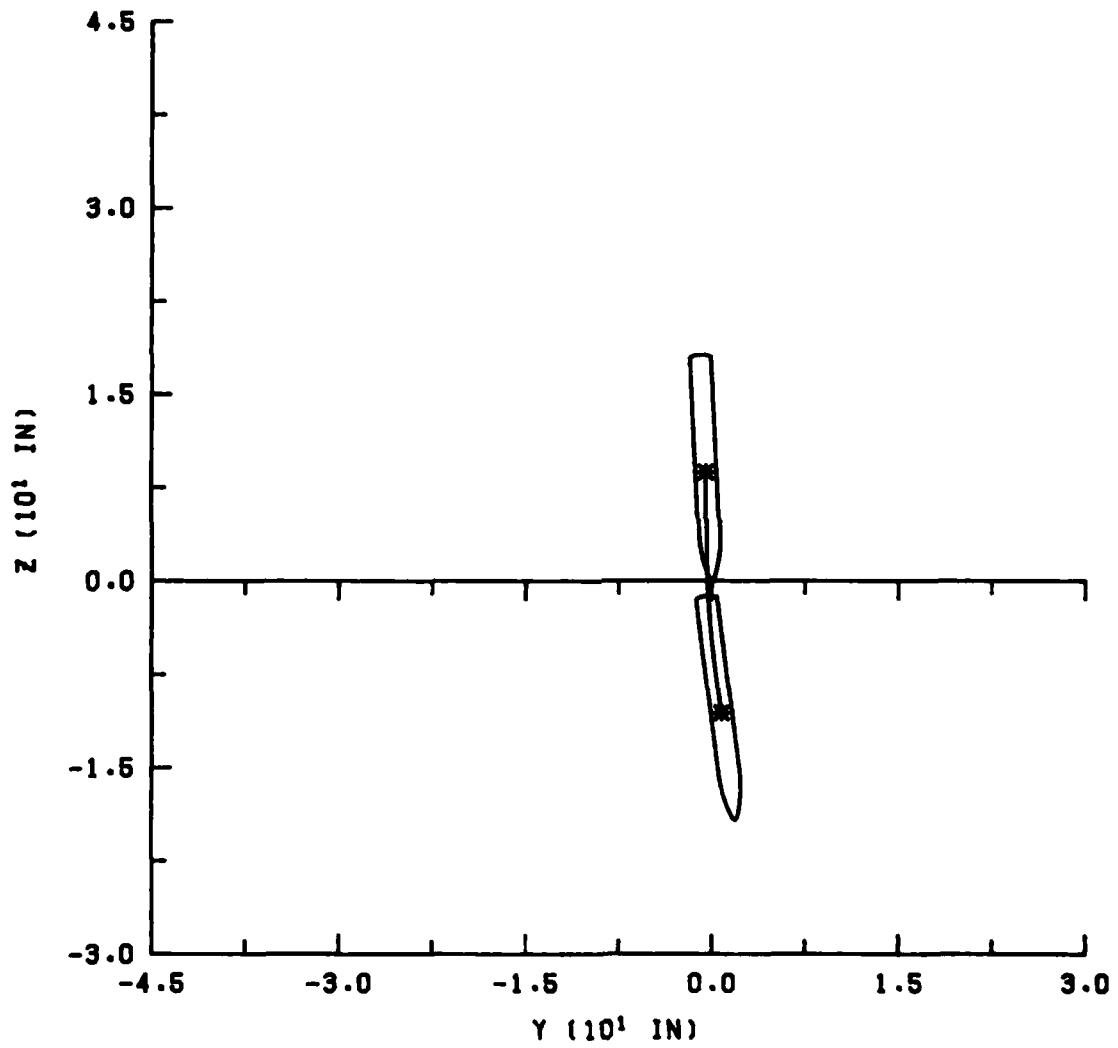


Figure 2.12 (Sheet 2 of 5).

SANDSTONE RBT SLED TEST...ALPHA=3 DEG
D=1.70 IN., W=9.48 LBS. V=1500 FPS
ALPHA=3.00 DEG., GAMMA=183 DEG.

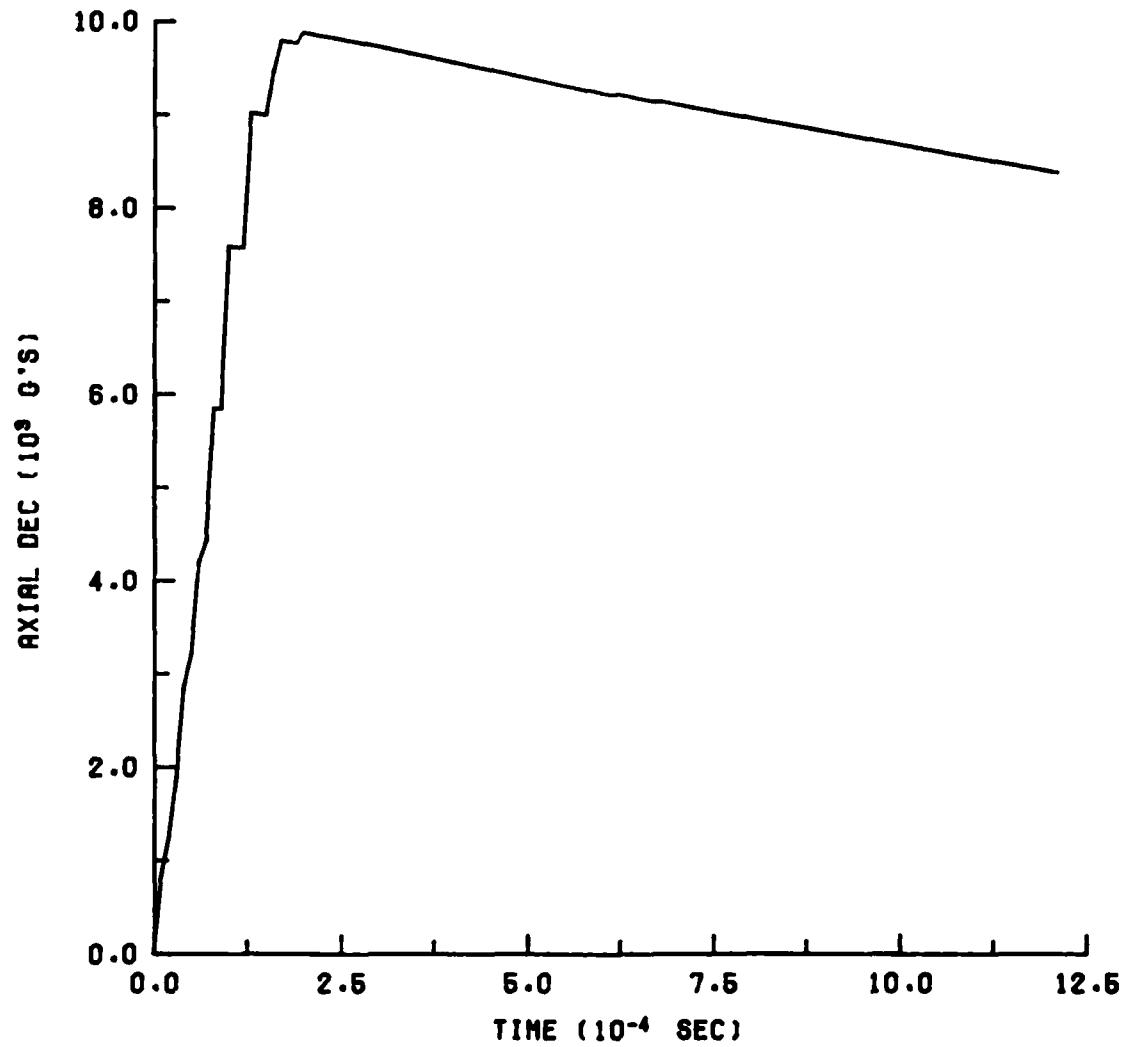


Figure 2.12 (Sheet 3 of 5).

SANDSTONE RBT SLED TEST...ALPHA=3 DEG
D=1.70 IN., W=9.48 LBS., V=1500 FPS
ALPHA=3.00 DEG., GAMMA=183 DEG.

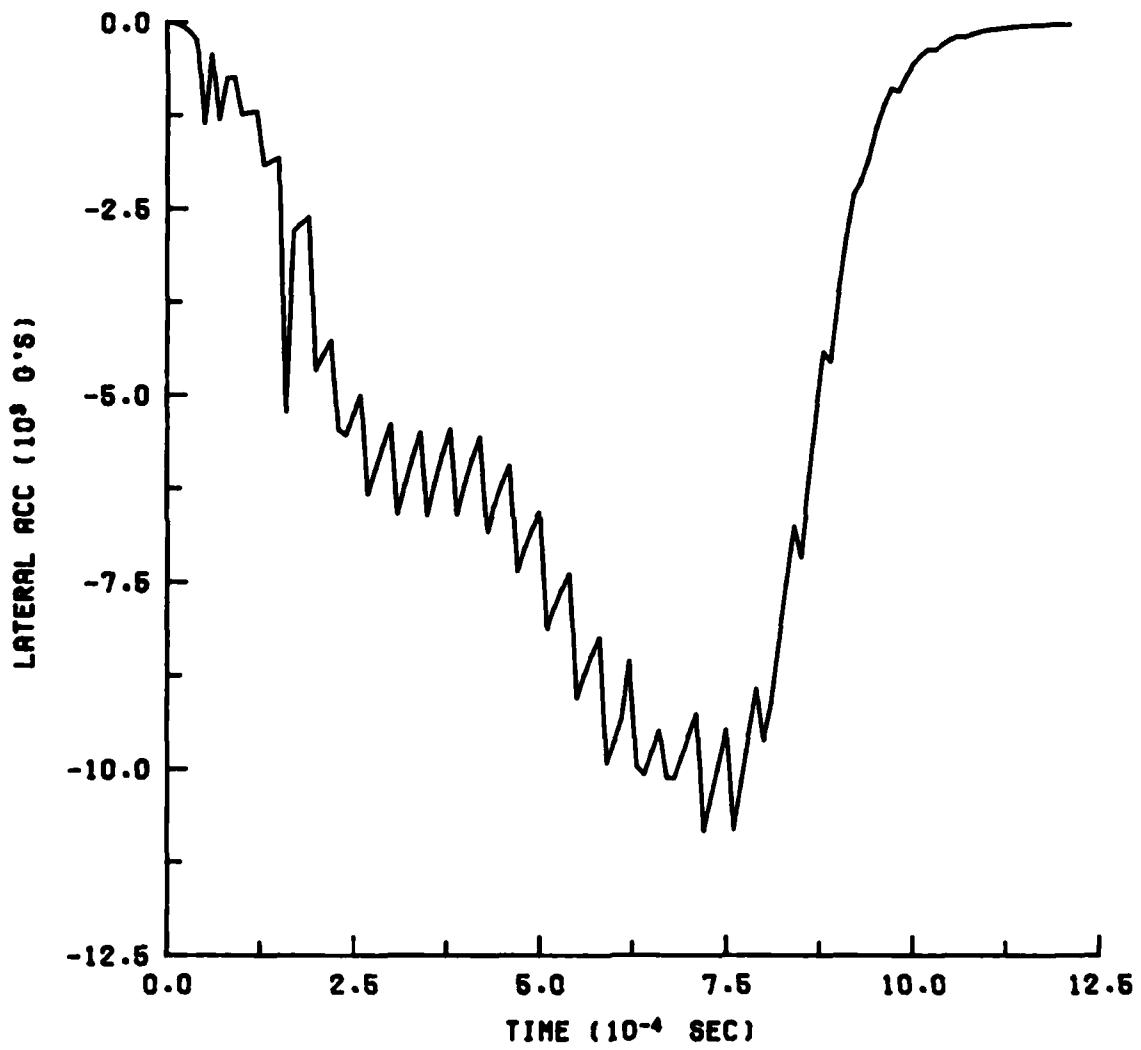


Figure 2.12 (Sheet 4 of 5).

SANDSTONE RBT SLED TEST...ALPHA=3 DEG
D=1.70 IN., W=9.48 LBS, V=1500 FPS
ALPHA=3.00 DEG., GAMMA=183 DEG.

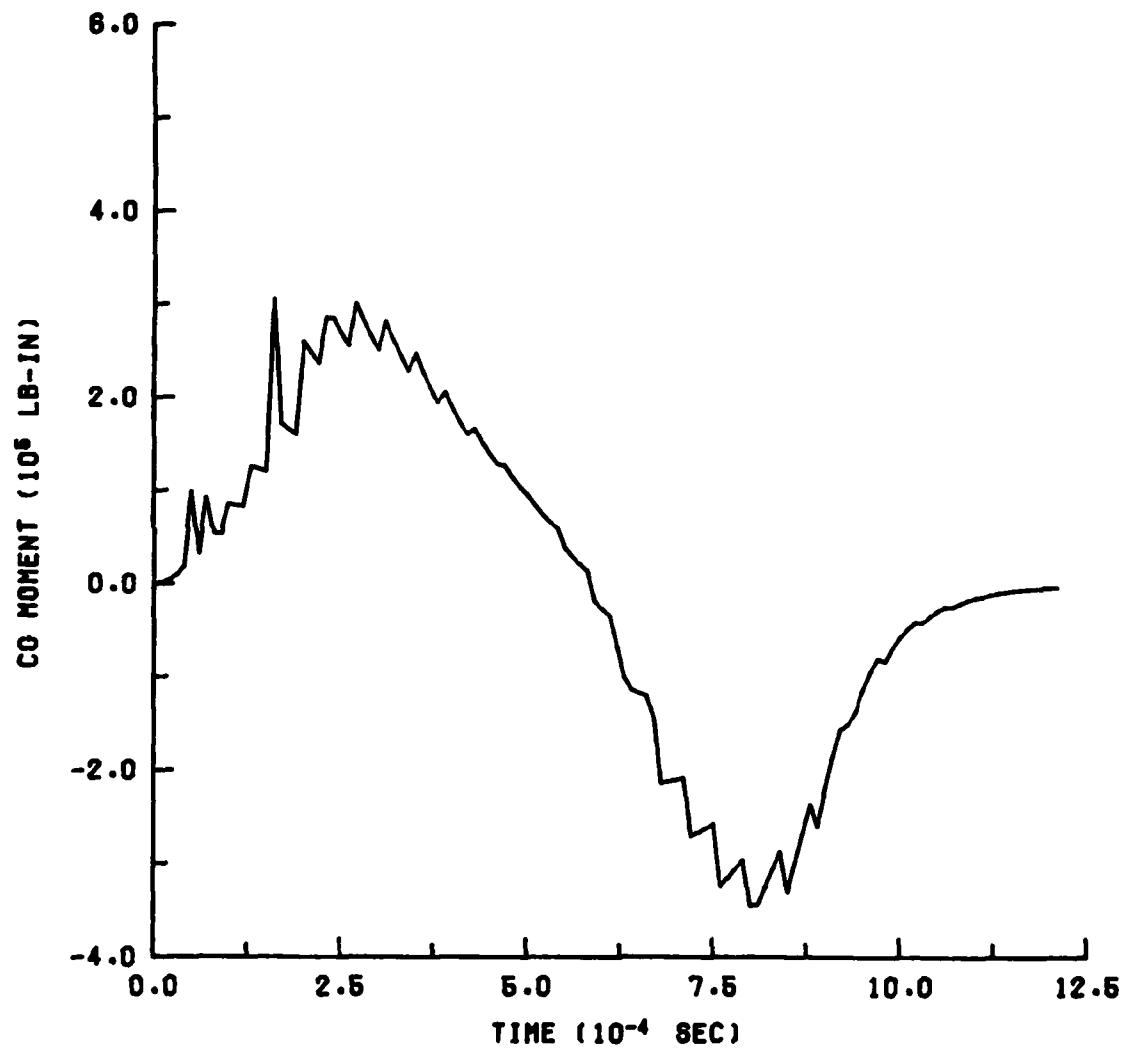


Figure 2.12 (Sheet 5 of 5).

REFERENCES

1. D. Henderson and R. L. Stephens; "Impact and Penetration Technology"; Paper Presentation, 28 November 1972; Fuze/Munitions Environmental Characterization Symposium, Picatinny Arsenal, Dover, NJ; Avco Corporation, Wilmington, MD.
2. C. W. Young; "Empirical Equations for Predicting Penetration Performance in Layered Earth Materials for Complex Penetrator Configurations"; Development Report No. SC-DR-72-0523, December 1972; Sandia Laboratories, Albuquerque, NM.
3. R. S. Bernard and D. C. Creighton; "Projectile Penetration in Soil and Rock: Analysis for Non-Normal Impact"; Technical Report SL-79-15, December 1979; U. S. Army Engineer Waterways Experiment Station, CE, Vicksburg, MS.
4. W. J. Patterson; Sandia Laboratories, Albuquerque, NM; untitled letter to P. F. Hadala, U. S. Army Engineer Waterways Experiment Station, CE, Vicksburg, MS, dated 16 December 1975.
5. W. R. Kampfe; "DNA 1/3-Scale·Pershing Penetrator Tests"; Letter Report to Defense Nuclear Agency dated October 1977; Sandia Laboratories, Albuquerque, NM.
6. R. S. Bernard and D. C. Creighton; "Non-Normal Impact and Penetration: Analysis for Hard Targets and Small Angles of Attack"; Technical Report S-78-14, September 1978; U. S. Army Engineer Waterways Experiment Station, CE, Vicksburg, MS.
7. W. J. Patterson; "DNA/Sandia Soil Penetration Experiment at DRES: Results and Analysis"; Report SAND 75-0001, October 1975; Sandia Laboratories, Albuquerque, NM.

APPENDIX A
INTERFACE AND FREE-SURFACE EFFECTS

The penetration resistance near a layer interface or free surface will usually differ from that of a purely homogeneous medium. The PENC02D code acknowledges the presence of such boundaries in a manner that is crude but qualitatively correct.

Each target layer is assigned a parameter I_T which loosely corresponds to the rigidity index (Reference 6) in tension. For soft materials (soil) the recommended value is $I_T = 10$, and for hard materials (rock and concrete) the recommended value is $I_T = 350$. In any case, there corresponds to I_T a damage radius r_D defined at each point on the projectile surface

$$r_D = r_p (I_T)^{1/3} \quad (A.1)$$

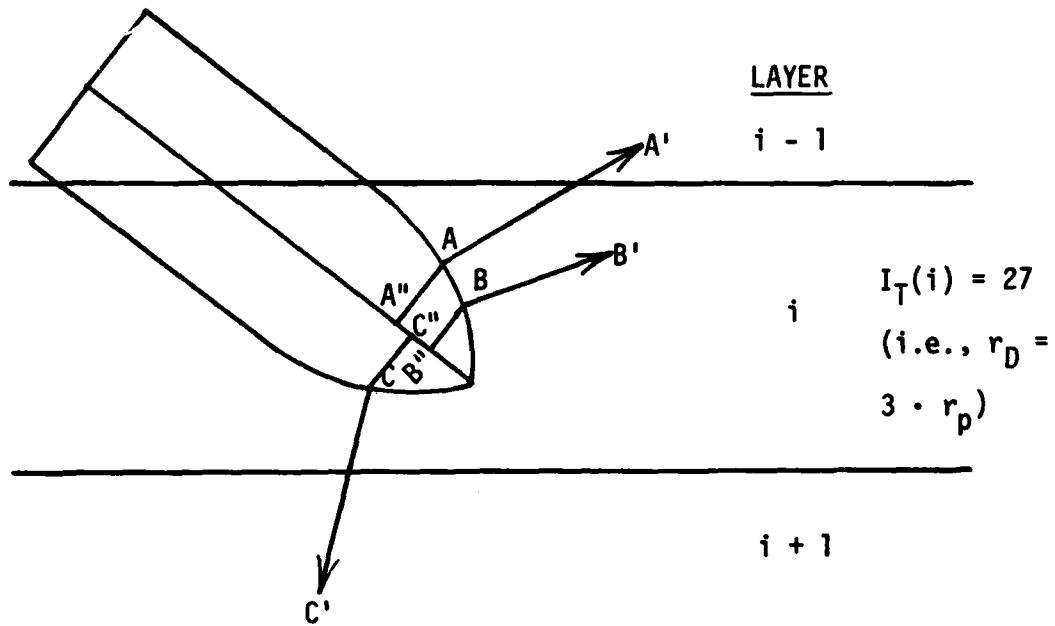
The quantity r_p represents the local cylindrical radius of the projectile. At each point on the projectile, a damage vector is defined such that

$$\vec{r}_D = \vec{n} r_p (I_T)^{1/3} \quad (A.2)$$

where \vec{n} is a unit vector normal to the projectile surface. The target properties to be used in defining the normal stress acting on any differential area on the surface of the projectile depend upon the particular layer in which the tip of the damage vector lies. In Figure A.1 the points A', B', and C' are the tips of damage vectors emanating from differential areas centered at points A, B, and C, respectively. If the tip is in the same layer occupied by the center of the projectile surface-area element (e.g., point B' in Figure A.1), the stress is calculated in the usual manner for that layer. But if the tip falls in an adjacent layer (e.g., points A' and C' in Figure A.1), the stress

calculation is made using the properties of the softer of the two layers.

The foregoing scenario represents an attempt to incorporate the effects of chipping and spallation into PENCO2D. For example, when a projectile penetrates a soft layer followed by a hard layer, the latter has no effect until contact is actually made. On the other hand, in the case of hard followed by soft, the soft layer begins to have an effect as the interface is approached. In general, whenever a projectile surface element is near an interface or free surface, there may be a reduction in the resisting stress if the adjacent layer is softer than the immediate layer.



Center of dA	Projectile Radius, r_p^*	Damage Vector, \vec{r}_D^{**}
A	$\ \text{line A-A''} \ $	$\vec{A-A'}$
B	$\ \text{line B-B''} \ $	$\vec{B-B'}$
C	$\ \text{line C-C''} \ $	$\vec{C-C'}$

* $\| a \|$ = magnitude of a .

** \vec{a} = vector a .

Figure A.1 Embedded projectile with damage vectors.

APPENDIX B

PENCO2D'S NEWTONIAN INTEGRATION WITH GUIDELINES TO RELATED INPUT

PENCO2D uses a two-dimensional (2D) Newtonian scheme to integrate the following equations of motion:

$$F_Y = m\ddot{Y} = F_y \cos \gamma - F_z \sin \gamma \quad (B.1)$$

$$F_Z = m\ddot{Z} = F_y \sin \gamma + F_z \cos \gamma \quad (B.2)$$

$$M = I\ddot{\gamma} = \int y dF_z - \int z dF_y \quad (B.3)$$

Equations B.1 through B.3 are Equations 2.9 through 2.11 from Reference 3, rewritten using PENCO2D's coordinate system. Figure B.1 shows the difference between the two coordinate systems. The integrals in Equation B.3 are evaluated over the projectile's entire surface area. Figure B.2 is a flowchart of the solution procedure used by PENCO2D. This integration begins by initializing Y , Z , and γ (the two spatial coordinates of the projectile center of gravity (CG) and the penetrator orientation) and their respective velocities, as well as time t . The current speed is calculated and used to compute a time step Δt according to the flowchart in Figure B.3. Subroutine FORCES is called to sum the stress components over the projectile's surface to obtain total force components F_Y and F_Z and total CG moment M . The changes in the translational and rotational velocities can then be calculated for the current time interval. New velocity components, as well as average velocities over the interval, are then determined so that the displacement component changes over the current interval can be computed. Once these components, along with the time, are updated and all stop conditions checked, the integration loop cycles to the next time interval. The problem continues until one of the inputted stop conditions is met.

The following parameters are responsible for adjusting the current time step to be used in the problem of interest: NUMNOS, FRAD, FANG, DTMIN, DTMAX, and IREDUC. Initially, NUMNOS is used to embed the penetrator nose. The nose is divided into NUMNOS equal-thickness sections, and each of the first NUMNOS time steps embeds one of these sections. If $S(t)$ is the path length traveled by the projectile up to time t , and L_N is the projectile nose length, then

$$S(t_i) - S(t_{i-1}) \approx \frac{L_N}{NUMNOS} \quad (B.4)$$

during the nose-embedding process. For a small interval Δt ,

$$V_i \Delta t_i \approx S(t_i) - S(t_{i-1}) \quad (B.5)$$

where V_i is the velocity at the beginning of the current interval. The first part of the penetration event is thus characterized by a time step given by

$$\Delta t_i \approx \frac{L_N}{V_i \cdot NUMNOS} \quad (\text{nose embedment}) \quad (B.6)$$

Generally, 10 to 50 is a reasonable range of values for NUMNOS, depending on target material, impact velocity, and nose length.

The time step used in the next portion of the penetration event is characterized using FRAD and FANG. FRAD is used to calculate a time step Δt that will translate the penetrator by a distance equal to FRAD projectile radii, while FANG is used to calculate a Δt that will rotate the penetrator by an angle equal to FANG radians. PENCO2D uses the smaller of the two time steps, allowing both rotational and translational limitations to be set. Some problems require that the rotation in a given time interval be limited (e.g., problems involving high angular accelerations) while other problems rely on translational motion

to determine time-step size. The time interval can be summarized as follows after nose embedment

$$\Delta t_i = \text{MIN} \left(\frac{\text{FRAD} \cdot r_o}{v_i}, \frac{\text{FANG}}{|\dot{y}_i|} \right) \quad (\text{B.7})$$

where r_o is projectile radius and v_i and \dot{y}_i are the current CG and angular velocities, respectively. How big should FRAD and FANG be? In multilayer problems, take the thinnest layer to be penetrated ($=T_{\min}$), estimate the number of time steps desired in this layer ($=N_D$), and calculate FRAD as

$$\text{FRAD} = \frac{T_{\min}}{N_D \cdot r_o} \quad (\text{multilayer problem}) \quad (\text{B.8})$$

In homogeneous problems, decide how many steps are required to accurately integrate the equation of motion; generally, $N_D = 100$ to 500 are required depending on target material, projectile size, and impact conditions. Let T_{\min} be an estimate of the total path length and use Equation B.8. This will give a starting value for FRAD. In all problems, two or more values of FRAD should be tried to check for convergence. If a problem is not dominated by rotation, FANG can be made large enough so that it will not affect the time interval calculation. Most of the current experience is with problems of relatively low angular motion. To select a value for FANG, estimate the maximum orientation change $|\Delta y_i|$ that can be tolerated during one time interval for the problem at hand. FANG is this value in radians.

When the velocity gets close to the final velocity, PENCO2D fixes the time step at a constant value to finish the calculation. Specifically, it uses the Δt of the increment just before v_i gets within $2\Delta v_i$ of the final velocity (VELF). IREDUC is used when the standard termination procedure is inappropriate, such as when significant lateral motion still remains at the end of a problem. If $\text{IREDUC} > 0$, then

IREDUC reductions, each by a factor of 3, will be made in both FRAD and FANG when the current CG velocity V_i has fallen to certain fractions of the impact velocity V_o , as given by Table B.1. If lateral motion has been cut off (Appendix D) before $V_i \leq 0.2V_o$, then these reductions do not occur.

Throughout the entire penetration event, the time step will never be less than DTMIN nor greater than DTMAX, regardless of the values of FRAD, FANG, etc. If for some reason a constant time step Δt is desired for a particular problem, just set

$$DTMIN = DTMAX = \Delta t \quad (\text{constant time step}) \quad (B.9)$$

Otherwise, DTMIN and DTMAX should be used to keep the time interval from getting too big or too small.

Table B.1 Velocities at which integration parameters FRAD and FANG are revised using time-step reduction.

Reduction Level	Velocity	New FRAD	New FANG
First	$0.2 \cdot V_o$	$FRAD_o/3^*$	$FANG_o/3^{**}$
Second	$0.1 \cdot V_o$	$FRAD_o/9$	$FANG_o/9$
Third	$0.05 \cdot V_o$	$FRAD_o/27$	$FANG_o/27$
Fourth	$0.025 \cdot V_o$	$FRAD_o/81$	$FANG_o/81$
etc.	etc.	etc.	etc.

* $FRAD_o$ = input value of FRAD.

** $FANG_o$ = input value of FANG.

PENCO2D Variable Explanation	PENCO2D	Reference 3
Projectile-fixed lateral direction, positive on clockwise-side of z-axis	y	$-x$
Projectile-fixed axial direction, positive forward of CG	z	z
Target-fixed horizontal direction, positive to the right	γ	x
Target-fixed vertical direction, positive upward	z	$-z$
Projectile orientation angle, measured from positive z-axis clockwise to upward vertical	γ	$\theta + \pi$

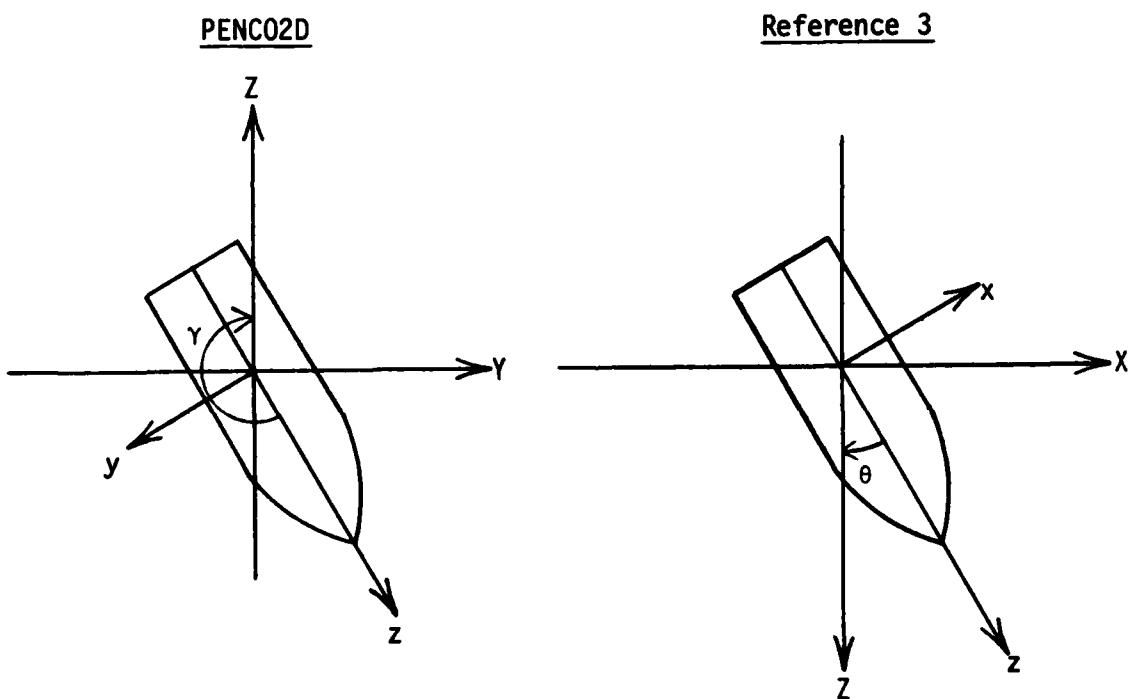


Figure B.1 Comparison of PENCO2D and Reference 3 coordinate systems.

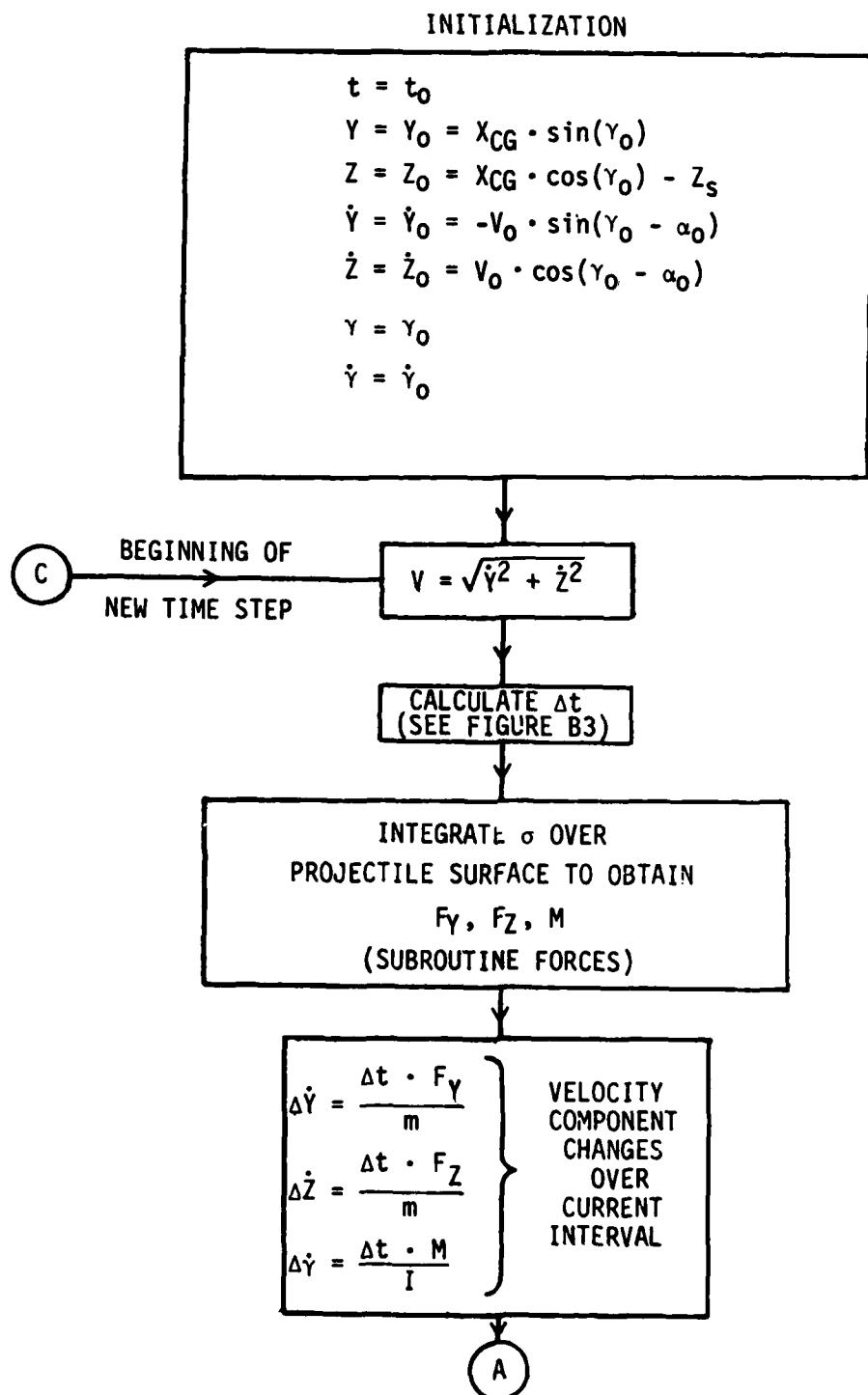


Figure B.2 Flowchart for PENC02D's Newtonian integration scheme (Sheet 1 of 3).

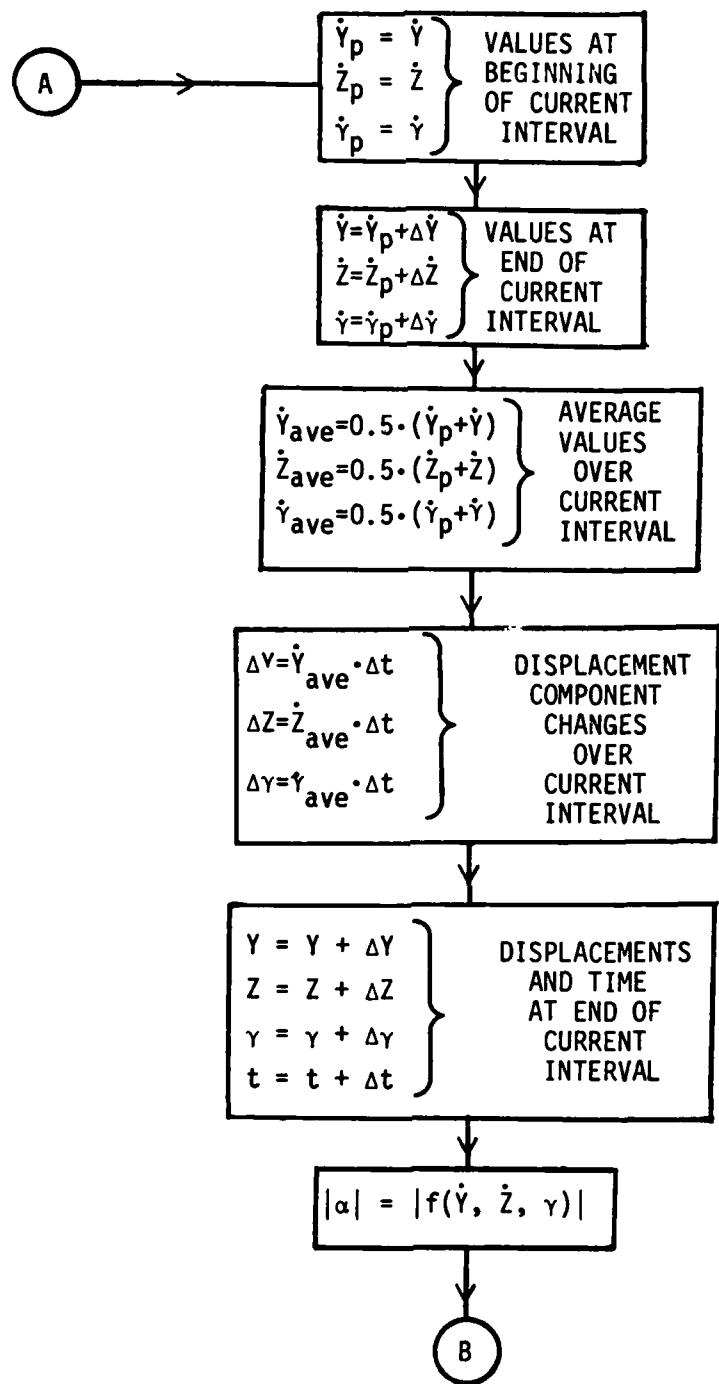


Figure B.2 (Sheet 2 of 3).

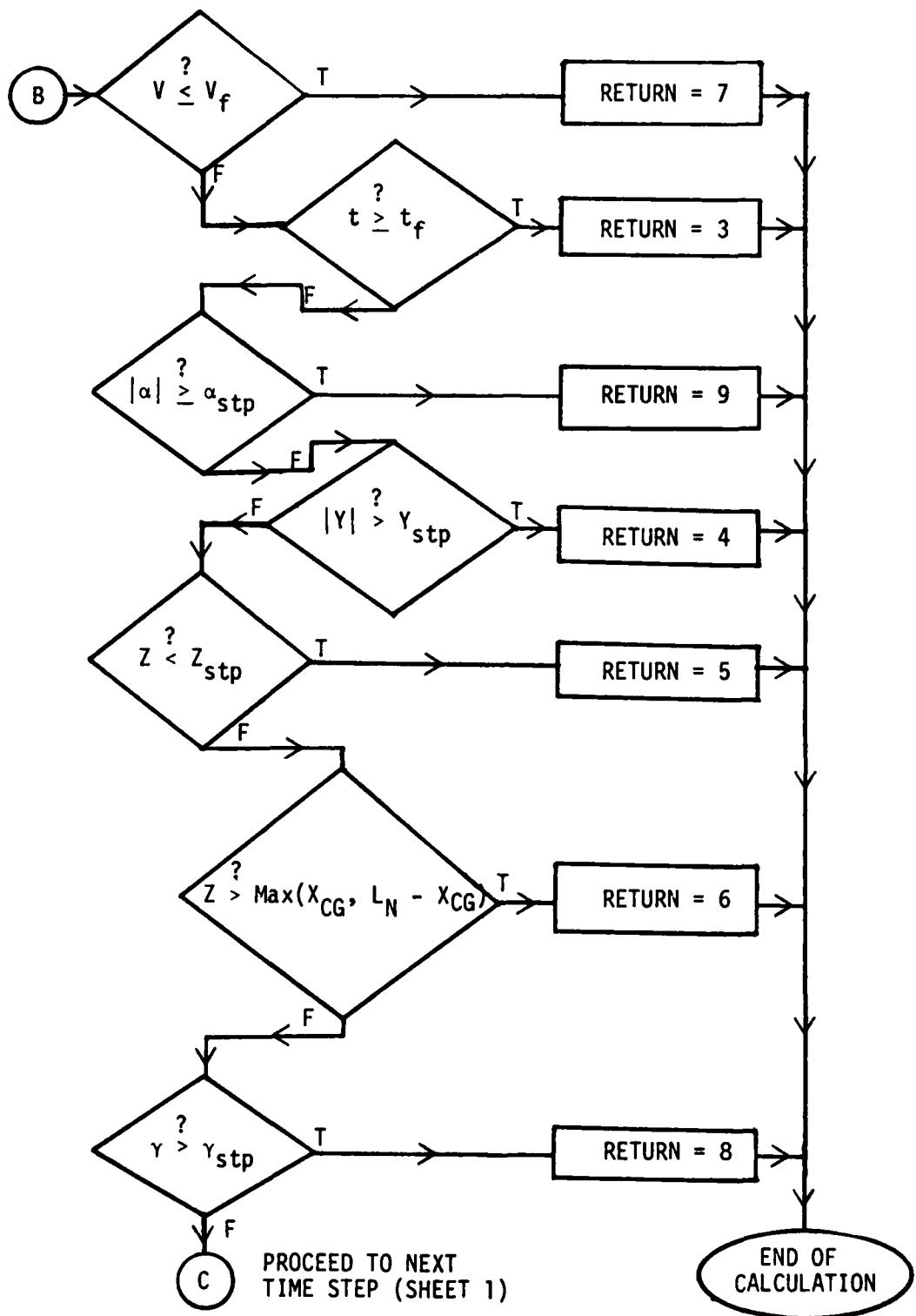


Figure B.2 (Sheet 3 of 3).

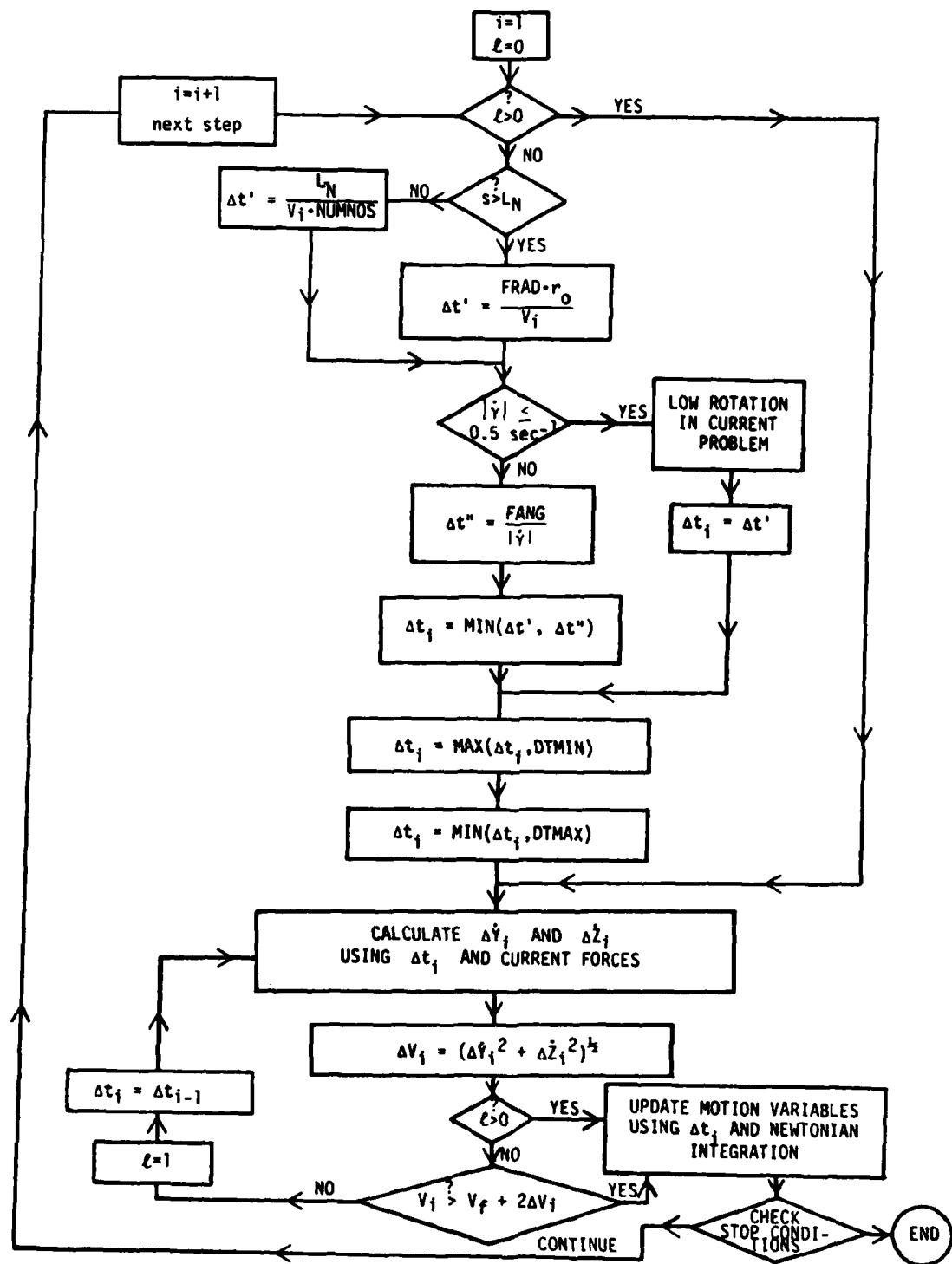


Figure B.3 Flowchart for determination of time step Δt .

APPENDIX C
GLOSSARY OF INPUT VARIABLES

ALPHAD	Initial angle of attack, deg
ALSTP	Problem ends if absolute value of angle of attack α ever exceeds ALSTP, deg
ARATIO	A differential area's length-to-width ratio; used by the geometry-generating routines if IOPSHP > 0, dimensionless
DALFAD	Used if NUMSTP > 1, to increment ALPHAD for subsequent run(s) using current data, deg
DELS(I)	Thickness of the i th longitudinal element, in
DENSITY(J)	Target mass density of j th layer; used for rock-like targets, $\frac{1\text{b}\cdot\text{s}^2}{4\text{ in}}$
DGAMAD	Used if NUMSTP > 1, to increment GAMMAD for subsequent run(s) using current data, deg
DTMAX	Maximum allowable time step, sec
DTMIN	Minimum allowable time step, sec
ELAS	Normal-stress smoothing parameter (Appendix E), rad
EJ	Secant ogive offset, in
FANG	Rotational time-step parameter (Appendix B), rad
FRAD	Translational time-step parameter (Appendix B), radii
FREKOT	Print output increment, seconds if NPRINT = 1, inches if NPRINT = 2
FREQI	Distance along trajectory path between successive scaled projectiles on automatically-generated trajectory plot, projectile lengths
GAMMAD	Initial projectile orientation measured from the line connecting the center of gravity (CG) to the nose tip clockwise around to vertical, deg

GAMSTP Problem ends if obliquity angle γ exceeds GAMSTP, deg

ICHO0Z(K) Selects the kth plot to be output according to this table:

ICHO0Z(K)	PLOT
1	CG axial acceleration versus time
2	CG axial deceleration versus time
3	CG lateral acceleration versus time
4	CG pitch moment versus time
5	CG velocity versus time
6	CG Y-displacement versus time
7	CG Z-displacement versus time
8	CG axial deceleration versus CG path length, (must be chosen last if chosen at all)

K = 1,2,...,NUMPLTS

[If NUMPLTS = 0, no plots are output except two that are automatically generated.]

IOPSSH Nose shape selection variable (Figure 2.2)
= 0 arbitrary shape (input R, THETAD, DELS arrays)
1 cone
2 cone with spherical tip
3 ogive with optional cone tip
4 blunted cone

IREDUC Number of times to reduce FRAD and FANG in order to complete a problem with significant lateral motion occurring late in the event (Appendix B)

NEL Number of longitudinal elements making up the projectile; an input quantity only if IOPSSH = 0

NEMOV2 Half the total number of circumferential elements comprising each longitudinal element; must be an even number (Figure 2.3)

NLAY	Number of target layers + 1 (layer 1 is the air layer above the target); must be $\leq 5^*$
NPRINT	Determines if FREKOT has units of time (=1) or path length traveled (=2)
NSTP	Maximum number of output steps
NUMNOS	Number of time steps to be used in nose embedment (see Appendix B)
NUMPLTS	Number of plots desired (i.e., using ICHOOZ-array to select specific plot types) in addition to the automatic projectile shape and trajectory plots; $0 \leq \text{NUMPLTS} \leq 8$
NUMSTP	Total number of runs to be generated with current set of data; subsequent runs use the same input except that DGAMAD and DALFAD are used to increment GAMMAD and ALPHAD, respectively
PHIMIN	Angle of approach ϕ_{\min} at which separation occurs (i.e., wake separation angle), deg (see Reference 3, Section 2.4)
R(I)	Average radius of the i th longitudinal element, in
RB	Radius of the conical frustum (IOPSHP = 4...blunted cone) or radius of conical tip (IOPSHP = 3...ogive nose with optional conical tip), in (see Figure 2.2 for projectile shapes)
RC	Length of spherical nose (IOPSHP = 2, see Figure 2.2), in
RJ	Ogive radius (IOPSHP = 3); if $RJ \geq 10^{-12}$, SN is computed; if $RJ < 10^{-12}$, SN is used to compute RJ, in
RN	Sphere radius (IOPSHP = 2), in
RO	Projectile radius r_o ; used in time-step computations (Appendix B), in geometry computations if IOPSHP $\neq 0$, and in plot heading, in

* If a problem requires more than four material layers (i.e., NLAY > 5), then a parameter variable (NMAX) in PENCO2D needs to be increased to the desired NLAY-value; the statements requiring change are the first statements in the main program and subroutines SEPARATE, FORCES, CALF, BACKWARD, and READIN (PENCO2D listing in Appendix G).

SN	Projectile nose length; necessary for time-step computation (Appendix B) and in geometry computations if IOPSHP \neq 0, in
SNUM(J)	Young's S-number (Reference 2) for the jth layer; used for soil targets; set = 0 for hard targets, dimensionless
THETAD(I)	Angle between the projectile axis and the line tangent to the surface of the ith longitudinal element, deg (Figure 2.3)
THETFD	Projectile aftbody flare angle if IOPSHP $>$ 0, deg
THETND	Cone half angle; used if IOPSHP = 1, 2, or 4, deg
TIMEF	Case ends when time $t \geq$ TIMEF, sec
TIMEI	Case begins at time $t =$ TIMEI, sec
TITLE	45-character (or less) title describing the current problem; printed out at the top of plots and printed output
VEL	Impact velocity, in/s
VELF	Case ends when CG velocity becomes less than or equal to VELF, in/s
W1I	Initial transverse rotation rate about CG, rad/s
WEIGHT*	Projectile weight, lb
XCG*	Location of projectile CG measured from the nose tip, in
XFA	Location of aft end of flare measured from CG; positive if forward of CG, in
XFF	Location of forward end of flare measured from CG (see XFA above), in
XICCG*	Projectile's transverse mass moment of inertia, lb-in ²

* Appendix F shows how to use a utility program called MOMENT to calculate CG location and moments of inertia for axisymmetric projectiles assuming either a material density or weight and an interior distribution of material(s).

XIRD(J)	Free-surface parameter I_T (see Appendix A) for jth layer, dimensionless
XLP	Projectile length, in
YIELD(J)	Yield strength for jth layer; used for hard targets; set = 0 for soil targets, lb/in^2
YSTOP	Case ends if $ Y $ (absolute value of horizontal coordinate of CG) $>$ YSTOP, in
ZM(J)	Distance from ground surface to the bottom of the jth layer (>0), in
ZSHIFT	Problem will begin with projectile nose tip located at absolute coordinates $Y = 0$ and $Z = -ZSHIFT$ (i.e., positive ZSHIFT embeds nose tip downward into target), in
ZSTOP	Case ends if Z (vertical coordinate of CG) $<$ ZSTOP, in

APPENDIX D
LATERAL CUTOFF

Stable projectiles may begin with significant obliquity and/or attack angle, but will eventually align themselves with the velocity vector. PENCO2D is set up to cut off the lateral motion (i.e., set $\alpha = 0$, $\dot{\gamma} = 0$) whenever this motion becomes insignificant. For this to occur, the following four conditions must exist simultaneously:

1. $|\alpha| \leq 0.01$ deg
2. $|\dot{\gamma}| \leq 0.05$ s^{-1}
3. $Z \leq Z_{cut}$ where Z_{cut} is a depth calculated to be out of range of free-surface effects (calculated internally in the code)
4. $\left| \frac{F_y}{W} \right| \leq 1.5$ g's where $\frac{F_y}{W} = \text{total lateral acceleration}$

When these conditions do occur, the motion becomes pure axial motion, indicated in the printout by PITCH RATE ($=\dot{\gamma}$), PITCH FORCE, and PITCH MOMENT all equalling zero, with α very small ($\leq 10^{-4}$ deg). (ALPHA is not exactly zero because of the computer round-off error.) The reasons for cutting off the lateral motion when α and $\dot{\gamma}$ are close to zero are twofold:

1. The projectile begins to oscillate between small positive and negative values of α , with the frequency increasing in time. These oscillations about $\alpha = 0$ do little to contribute to the overall motion of the projectile but do a lot to disrupt the integration scheme.
2. Pure axial motion is much faster to integrate because a number of calculations can be skipped once lateral motion is cut off completely.

APPENDIX E
USE OF ELAS TO SMOOTH NORMAL STRESS CURVE

As $|\alpha|$ approaches 0, the local normal velocity component v_n for the differential-area elements (dA's) on the windward side of the projectile also approaches 0 for elements whose surface is parallel to the projectile axis (i.e., aftbody elements on nonflared projectiles). There is still a significant tangential component, however, as long as the projectile still translates with some velocity. Thus, the angle

$$\xi = \sin^{-1} \left(\frac{v_n}{v} \right) \quad (E.1)$$

approaches 0 (i.e., $v_n \rightarrow 0$ but v does not). This creates a numerical problem with the normal stress σ , because the σ versus ξ curve is steep near $\xi = 0$. In fact, the slope becomes vertical as $\xi \rightarrow 0$.

To help alleviate the problem, a smoothing parameter ELAS is introduced. ELAS defines a value of ξ below which smoothing will take place. The region $0 \leq \xi \leq \text{ELAS}$ of the $\sigma - \xi$ curve is replaced with a curve defined by

$$\sigma^*(\xi) = \sigma(\xi) + \sin \left[\frac{\xi \cdot \pi}{2 \cdot \text{ELAS}} \right] \quad (E.2)$$

where $\sigma(\xi)$ is the original σ -function for normal stress prior to smoothing. Figure E.1 shows a typical $\sigma - \xi$ graph in the region near $\xi = 0$ for a 6-inch projectile penetrating a soil target ($S = 5$) with $v = 0.5V_o$, V_o = impact velocity = 500 ft/s. The solid line shows the unaltered curve. The dashed curve is the PENCO2D version for the region $0 \leq \xi \leq \text{ELAS}$, demonstrating a smoother, gentler gradient than the original curve. The obvious problem with this procedure is the reduction in lateral force experienced by those differential elements with small values of v_n . Thus, it is desirable to choose ELAS small enough

so that the total lateral force is not appreciably reduced. The minimum value of ELAS depends upon the gradient of the $\sigma - \xi$ curve and the current time step. If the value of ELAS is too small, one time increment may be enough to allow ξ to jump over the smoothing section, thus allowing rapid oscillations (spikes) to occur in the lateral acceleration-time history. The time-step parameters FRAD and FANG, as well as ELAS, must be chosen so as to control spiking without significantly reducing the peak lateral acceleration. If ELAS is too big, the lateral acceleration will be attenuated. If both spiking and lateral peak reduction occur, a smaller time step is needed. Ideally, one should use as large an ELAS value as possible without causing attenuation.

To find this maximum allowable ELAS, call it $ELAS_{max}$, two or more runs should be executed with different ELAS values so that the lateral acceleration-time curves can be compared. Figure E.2 relates three runs made for the problem examined in Figure E.1. The solid curve shows a numerical spike at point C; thus, ELAS used for that run must be too small. The alternating dash-dot curve shows reduced peak amplitudes at points A and B; ELAS used for that run is therefore too large (i.e., $ELAS > ELAS_{max}$). After a few runs, an acceptable $ELAS \approx ELAS_{max}$ can be converged upon. In Figure E.2, the dashed line for $ELAS = 0.01$ exhibits a very slight peak reduction with no numerical spiking. The conclusion is that $ELAS_{max} \approx 0.01$, and experience with PENCO2D has shown that $ELAS = 0.01$ is a good starting value for most materials. Nevertheless, several values of ELAS should be tried for any new problem using PENCO2D.

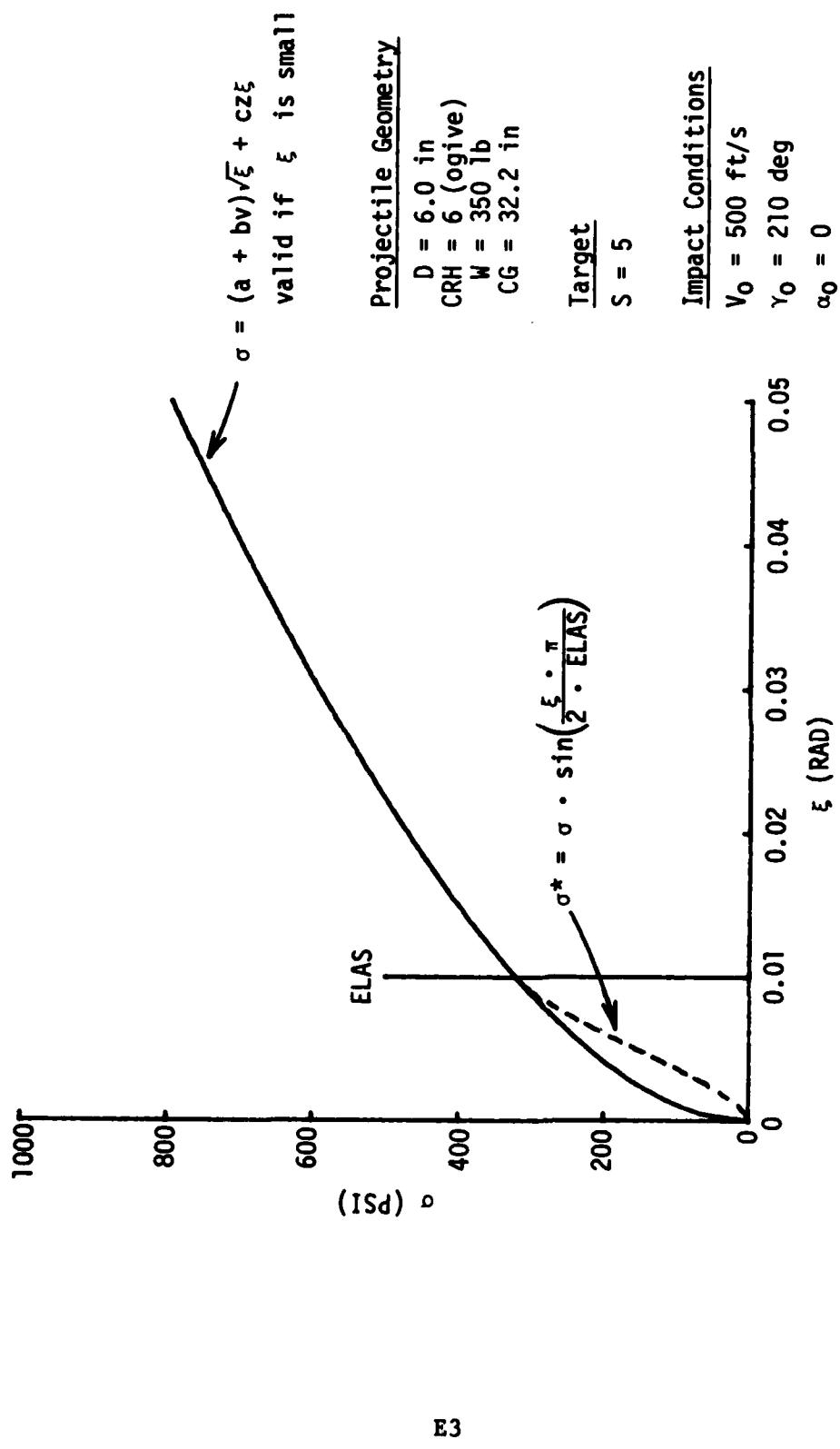


Figure E.1 Normal-stress smoothing achieved using ELAS for 6-inch projectile obliquely impacting soil; curve shown for $v = 0.5V_0$ and $z = 11$ feet.

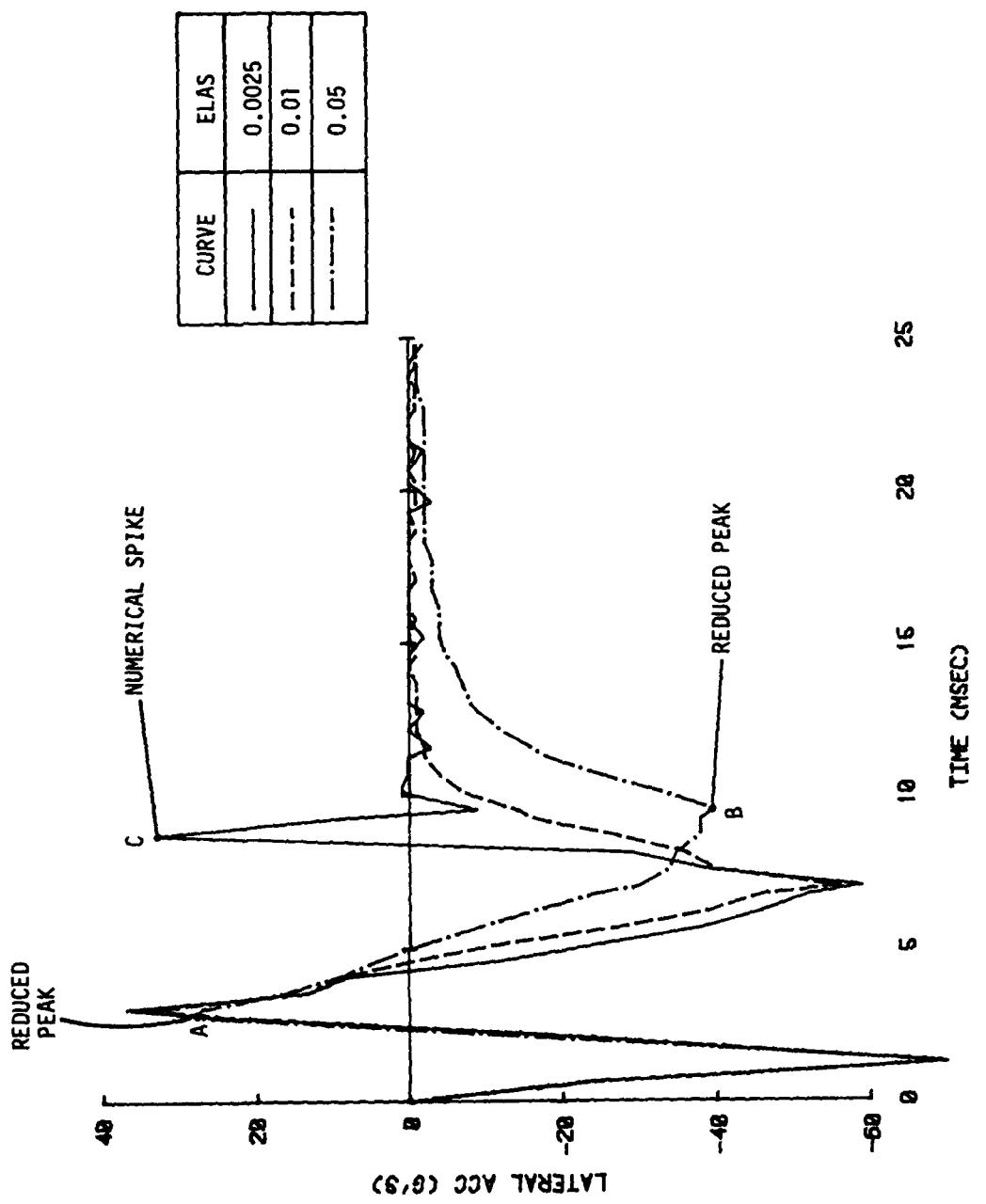


Figure E.2 Lateral acceleration-time histories for different values of ELAS; oblique impact into soil.

APPENDIX F

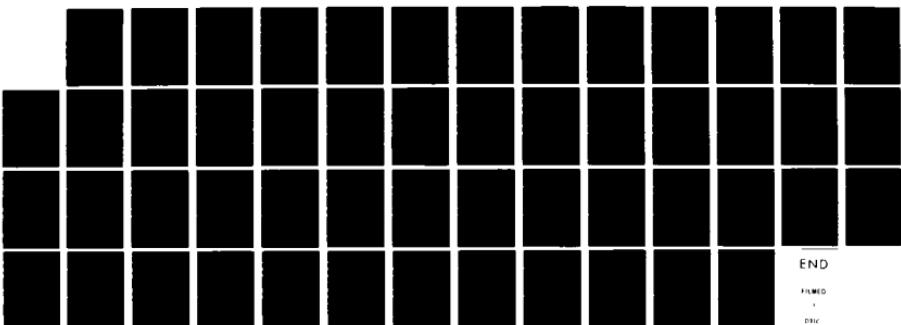
MOMENT: UTILITY PROGRAM TO CALCULATE WEIGHT, CENTER-OF-GRAVITY LOCATION, AND MOMENTS OF INERTIA FOR AXISYMMETRIC PROJECTILES

When preparing input for a PENCO2D run, it may be necessary to calculate some or all of the projectile parameters. This appendix presents a utility computer program called MOMENT that calculates the weight, center-of-gravity (CG) location, and polar and transverse mass moments of inertia for any axisymmetric penetrator for which the weight (or density), shape, and location of each interior section are known. The better the detail of the interior layout, the better the accuracy of MOMENT's resulting output will be. Two examples will be set up to demonstrate use of program MOMENT.

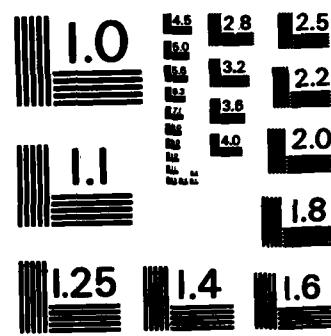
A listing of MOMENT is given in Figure F.1. This program is set up to run on a Honeywell DPS-1 time-sharing system. The first line in the listing locates additional files to be compiled with MOMENT at the time of execution; these files contain the plot subroutines PLOTS, PLOT, SCALIT, SYMBOL, AXIS13, and NUMBER. All are routines used by subroutine PLOTIT to scale the data automatically, then draw the resulting penetrator with labeled axes on a Tektronix 4662 plotter connected to a time-sharing terminal. There is a glossary of input quantities at the beginning of the listing explaining input variables and the appropriate units.

Program MOMENT requires one input file to be set up prior to execution, containing weights (or densities) and geometry information. Figure F.2 lists the general form of this file. Generally, a projectile is divided into one or more "figures" for which the weight or mass density is known. Each figure must then be divided into right-circular conical frustum "sections" as shown by the shaded area in the inset in Figure F.2. For each figure, the number of conical sections (NSECT) must be specified as well as a number (IDENS) that indicates whether the weight (IDENS = 1) or density (IDENS = 2) is to be input. The next line contains the weight (or density) of the figure. The following NSECT

AD-A121 890 NON-NORMAL PROJECTILE PENETRATION IN SOIL AND ROCK: 2/2
USER'S GUIDE FOR COMP. (U) ARMY ENGINEER WATERWAYS
EXPERIMENT STATION VICKSBURG MS STRUC. D C CREIGHTON
UNCLASSIFIED SEP 82 WES/TR/SL-82-7 F/G 19/4 NL



END
FILED
DRC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS - 1963 - A

lines contain the dimensions of each section in that figure. This information is repeated for each figure until the entire projectile has been described.

The first example is a blunted conical-nosed projectile (Figure F.3) made of one material (D6AC steel) with a known mass density of 15.23 slugs/ft³. Only one three-section figure is needed to describe the entire projectile exactly, and Figure F.4 shows the resulting input file for this problem. Line 001 indicates the total number of figures to be used (=1). Line 002 shows the number of sections (=3) and indicates that the density of the figure will be input (IDENS = 2). Line 003 is the mass density (=15.23 slugs/ft³). Lines 004 through 006 give the required geometry of all three sections as outlined in Figure F.2. Figure F.5 gives the DPS-1 time-sharing question-answer sequence including the desired geometric quantities printed out. (User responses in Figure F.5 are underlined.) Figure F.6 shows the code-generated plot verifying that the calculated results are indeed for the projectile of interest beyond all reasonable doubt. The plot shows the sectioning of the projectile with the little number 1's indicating the figure number to which the section(s) belong. The large "*" shows the resulting CG location.

In July 1974, Sandia Laboratories conducted a series of deep penetration tests on the Watching Hill Blast Range near Ralston, Alberta, Canada. These experiments, which were sponsored by the Defense Nuclear Agency, are documented by Patterson (Reference 7). One of the penetrators used at Watching Hill is schematized in Figure F.7. Because weights of individual "compartments" are known rather than densities, the projectile can be divided into the eight figures indicated in Figure F.7, and the IDENS = 1 option used. The resulting input file is shown in Figure F.8 and is self-explanatory with the help of the first half of sheet 1 of Figure F.1, to explain the terminology, and Figure F.2 to show the layout of the input file. Figure F.9 lists the time-sharing printout, with user responses underlined, that results from executing MOMENT. Finally, Figure F.10 plots the resulting form and gives the

user an idea as to how good his geometrical approximations were. The small numerals in the projectile sections of Figure F.10 indicate the figure to which the section in question belongs; only the larger sections receive numbers to enhance readability.

MOMENT
UTILITY PROGRAM FOR WEIGHT, CG, AND MOMENT-OF-INERTIA COMPUTATION

```

0010*#RUN *;ROSDS466/CPLOTS,E;SCALE,E
0020C THIS PROGRAM CALCULATES CG LOCATION AND MOMENT OF INERTIA FOR
0030C ANY AXISYMMETRIC PROJECTILE FOR WHICH DENSITIES OR WEIGHTS OF
0040C EACH INTERNAL COMPONENT IS KNOWN. THE PERTINENT VARIABLES
0050C ARE:
0060C
0070C NFIGS.....NUMBER OF (DIFFERENT-DENSITY) FIGURES
0080C (I.E. SECTIONS) WITHIN THE PROJECTILE.
0090C NSECT(I),I=1,NFIGS.....NUMBER OF SUBSECTIONS INTO WHICH EACH
0100C FIGURE IS DIVIDED. EACH SUBSECTION
0110C IS A RIGHT-CIRCULAR CONICAL FRUSTRUM
0120C OUTSIDE AND INSIDE (I.E. INSIDE CAN
0130C BE A HOLLOW CONICAL SECTION).
0140C IDENS(I),I=1,NFIGS....=1 TO INPUT FIGURE'S WEIGHT
0150C 2 TO INPUT FIGURE'S MASS DENSITY
0160C W(I).....WEIGHT OF I'TH FIGURE (LBS).
0170C DENSITY(I).....MASS DENSITY OF I'TH FIGURE (SLUGS/
0180C CU.FT.)
0190C FOR EACH SUBSECTION:
0200C R2B(J).....OUTSIDE FORWARD RADIUS OF J'TH SUBSECTION.
0210C R2S(J).....INSIDE FORWARD RADIUS OF J'TH SUBSECTION.
0220C R1B(J).....OUTSIDE AFT RADIUS OF J'TH SUBSECTION.
0230C R1S(J).....INSIDE AFT RADIUS OF J'TH SUBSECTION.
0240C Y1(J).....DISTANCE FROM NOSE TIP TO FORWARD END
0250C OF J'TH SUBSECTION.
0260C DX(J).....THICKNESS OF J'TH SUBSECTION.
0270C NOTE: J=1,NSECT(I)...I'TH FIGURE...ALL UNITS ABOVE IN INCHES
0280C UNLESS OTHERWISE STATED.
0290C
0300 COMMON/A1/R2B(50),R2S(50),R1B(50),R1S(50),Y1(50),
0310 &Y2(50),NTOTSEC,ZCG,NSECT(10),DX(50)
0320 DIMENSION TVOL(10),IDENS(10),W(10),DENSITY(10),
0330 &XMASS(10),
0340 &E(50),F(50),G(50),H(50),VOL(50)
0350 &,A(50),B(50),C(50),D(50),X1(50),X2(50),MTP(50),MTT(50)
0360 LOGICAL LDENS(10)
0370 CHARACTER PRTX1
0380 REAL MY,MZ,MTP,MTT
0390 CHARACTER X8XINPUT(2),PRTX1
0400 DATA XINPUT(2)/1H;/
0410 PRINT,"INPUT FILENAME"
0420 READ,XINPUT(1)
0430 PI=3.1416; GRAV=32.2
0440 CALL ATTACH(1,XINPUT,3,0,,)
0450 READ(1,75) LINE,NFIGS
0460 FORMAT(V)
0470 NREAD=0
0480 DO 10 I=1,NFIGS
0490 TVOL(I)=0.
0500 READ(1,75) LINE,NSECT(I),IDENS(I)

```

75

Figure F.1 Listing of program MOMENT (Sheet 1 of 5).

MOMENT
UTILITY PROGRAM FOR WEIGHT, CG, AND MOMENT-OF-INERTIA COMPUTATION

```

0510      IF(IDENS(I).EQ.2) GO TO 11
0520      READ(1,75) LINE,W(I)
0530      LDENS(I)=.FALSE.
0540      GOTO 12
0550 11      READ(1,75) LINE,DENSITY(I)
0560      LDENS(I)=.TRUE.
0570 12      IF(I.EQ.1) GO TO 13
0580      NREAD=NREAD+NSECT(I-1)
0590 13      CONTINUE
0600      DO 14 J=NREAD+1,NREAD+NSECT(I)
0610      READ(1,75) LINE,R2B(J),R2S(J),R1B(J),R1S(J),Y1(J),DX(J)
0620      Y2(J)=Y1(J)+DX(J)
0630      E(J)=R1B(J)-R2B(J)
0640      F(J)=R2B(J)*Y2(J)-R1B(J)*Y1(J)
0650      G(J)=R1S(J)-R2S(J)
0660      H(J)=R2S(J)*Y2(J)-R1S(J)*Y1(J)
0670      VOL(J)=(PI/DX(J)**2)*((Y2(J)**3-Y1(J)**3)*(E(J)**2-G(J)**2)
0680      &/3.+(Y2(J)**2-Y1(J)**2)*(E(J)*F(J)-G(J)*H(J))+DX(J)*
0690      &(F(J)**2-H(J)**2))/(12.***3)
0700 14      TVOL(I)=TVOL(I)+VOL(J)
0710 10      CONTINUE
0720      NTOTSEC=J
0730      DO 20 I=1,NFIGS
0740      IF(LDENS(I)) GO TO 20
0750      XMASS(I)=W(I)/GRAV
0760      DENSITY(I)=XMASS(I)/TVOL(I)
0770 20      CONTINUE
0780      TMASS=0.; TZCG=0.; MN=0
0790      DO 30 J=1,NFIGS
0800      IF(J.EQ.1) GO TO 23
0810      MN=MN+NSECT(J-1)
0820 23      N=NSECT(J)
0830      DO 25 K=1+MN,N+MN
0840      XMULT=PI*DENSITY(J)/(DX(K)**2)
0850      DZCG=((Y2(K)**4-Y1(K)**4)*(E(K)**2-G(K)**2)/4. +
0860      &(Y2(K)**3-Y1(K)**3)*(2*E(K)*F(K)-2*G(K)*H(K))/3. +
0870      &(Y2(K)**2-Y1(K)**2)*(F(K)**2-H(K)**2)/2.)/(12.***3)
0880      DZCG=DZCG*XMULT
0890      DMASS=DENSITY(J)*VOL(K)
0900      TZCG=TZCG + DZCG
0910 25      TMASS=TMASS + DMASS
0920 30      CONTINUE
0930      ZCG=TZCG/TMASS
0940      TWEIGHT= TMASS * GRAV
0950      PRINT,
0960      PRINT,"DO YOU WANT TO CALCULATE MOMENTS"
0970      READ,PRT
0980      IF(PRT.EQ.1)N) GO TO 999
0990      DO 40 I=1,NFIGS
1000      DENPI=PI*DENSITY(I)/(12.***3)

```

Figure F.1 (Sheet 2 of 5).

MOMENT
UTILITY PROGRAM FOR WEIGHT, CG, AND MOMENT-OF-INERTIA COMPUTATION

```

1810 IF(I.EQ.1) GO TO 43
1820 NREAD=NREAD+NSECT(I-1)
1830 GO TO 44
1840 43 NREAD=0; MZ=0.; MY=0.
1850 44 CONTINUE
1860 DO 45 J=NREAD+1,NREAD+NSECT(I)
1870 X2(J)=ZCG - Y1(J)
1880 X1(J)=ZCG - Y2(J)
1890 A(J)=-E(J)
1900 B(J)=R1B(J)*X2(J) - R2B(J)*X1(J)
1910 C(J)=-G(J)
1920 D(J)=R1S(J)*X2(J) - R2S(J)*X1(J)
1930 ONE=R2B(J)**4 + R2B(J)**3*R1B(J) + (R1B(J)*R2B(J))**2
1940 & + R2B(J)*R1B(J)**3 + R1B(J)**4
1950 TWO=R2S(J)**4 + R2S(J)**3*R1S(J) + (R2S(J)*R1S(J))**2
1960 & + R2S(J)*R1S(J)**3 + R1S(J)**4
1970 MTP(J)=DENPI*DX(J)*(ONE-TWO)/10.
1980 FIVE=(X2(J)**5-X1(J)**5)*(A(J)**2-C(J)**2)/5.
1990 SIX=(X2(J)**4-X1(J)**4)*(A(J)*B(J)-C(J)*D(J))/2.
2000 SEVEN=(X2(J)**3-X1(J)**3)*(B(J)**2-D(J)**2)/3.
2010 ADD=DENPI*(FIVE+SIX+SEVEN)/(DX(J)**2)
2020 MTT(J)=.5*MTP(J)+ADD
2030 MY=MY+MTT(J)
2040 45 MZ=MZ+MTP(J)
2050 40 CONTINUE
2060 PRINT,; PRINT,
2070 PRINT 400,MY/144.,MY*GRAV,MY*1.355668/144.
2080 PRINT,
2090 PRINT 401,MZ/144.,MZ*GRAV,MZ*1.355668/144.
2100 999 PRINT,
2110 PRINT 402,TWEIGHT,TWEIGHT*.4536
2120 PRINT,
2130 PRINT 403,ZCG,ZCG*2.54
2140 PRINT,"DO YOU WANT TO PLOT"
2150 READ,PRT
2160 IF(PRT.EQ.1HY) CALL PLOTIT
2170 400 FORMAT(2X,"TOTAL TRANSVERSE MOMENT OF INERTIA =",1PE12.4,
2180 &1X,"SLUG-FEET**2",/,37X,"=",E12.4,1X,"LB.-INCHES**2",/,
2190 &37X,"=",E12.4,1X,"KG.-METERS**2")
2200 401 FORMAT(2X,"TOTAL",2X,"POLAR",2X,"MOMENT",2X,"OF INERTIA",
2210 &2X,"=",1PE12.4,1X,"SLUG-FEET**2",/,37X,"=",E12.4,1X,"KG.-METERS**2")
2220 402 FORMAT(2X,"TOTAL PROJECTILE WEIGHT =",1PE12.4,1X,"LBS.",/,26X,"=",E12.4,1X,"KG.")
2230 403 FORMAT(2X,"CENTER OF GRAVITY LOCATED",0PF10.4,2X,
2240 &"INCHES FROM NOSE TIP",/,27X,"(",F9.4,2X,"CM.")")
2250 STOP
2260 END

```

Figure F.1 (Sheet 3 of 5).

MOMENT
UTILITY PROGRAM FOR WEIGHT, CG, AND MOMENT-OF-INERTIA COMPUTATION

```

1490      SUBROUTINE PLOTIT
1500      COMMON/A1/R2B(50),R2S(50),R1B(50),R1S(50),
1510      &Y1(50),Y2(50),NTOTSEC,ZCG,NSECT(10),DX(50)
1520      DIMENSION SX(4)
1530      CALL PLOTS("A")
1540      CALL PLOT(2.,4.25,-3)
1550      SX(1)=0.
1560      SX(2)=0.
1570      RMAX=0.
1580      DO 1 J=1,NTOTSEC
1590      SX(2)=MAX(SX(2),Y2(J))
1600      RMAX=MAX(RMAX,R1B(J))
1610      1  CONTINUE
1620      CALL SCALIT(SX,7.,2,1)
1630      SFX=SX(4)
1640      SX(2)=RMAX
1650      CALL SCALIT(SX,3.,2,1)
1660      SFY=SX(4)
1670      SFACT=MAX(SFX,SFY)
1680      IF(IFIX(IFIX(SFACT)/SFACT)) 6,6,5
1690      6  ND=1
1700      GO TO 7
1710      5  ND=-1
1720      7  CONTINUE
1730      SFINV=1./SFACT
1740      H=0.25
1750      CALL SYMBOL(ZCG*SFINV,0.,H,11.0.,-1)
1760      FIRSTY=-3*SFX
1770      CALL AXIS13(0.,-3.,"INCHES",-6.,14.7.,ND,0,0.,SFACT,1.,1,0)
1780      CALL AXIS13(0.,-3.,"INCHES",6.,14.6.,ND,1,FIRSTY,SFACT,1.,1,0)
1790      NF=1
1800      N=NSECT(1)
1810      DO 3 J=1,NTOTSEC
1820      CALL PLOT(Y1(J)*SFINV,-R2S(J)*SFINV,3)
1830      CALL PLOT(Y1(J)*SFINV,-R2B(J)*SFINV,2)
1840      CALL PLOT(Y2(J)*SFINV,-R1B(J)*SFINV,2)
1850      CALL PLOT(Y2(J)*SFINV,-R1S(J)*SFINV,2)
1860      CALL PLOT(Y1(J)*SFINV,-R2S(J)*SFINV,2)
1870      CALL PLOT(Y1(J)*SFINV,R2S(J)*SFINV,3)
1880      CALL PLOT(Y1(J)*SFINV,R2B(J)*SFINV,2)
1890      CALL PLOT(Y2(J)*SFINV,R1B(J)*SFINV,2)
1900      CALL PLOT(Y2(J)*SFINV,R1S(J)*SFINV,2)
1910      CALL PLOT(Y1(J)*SFINV,R2S(J)*SFINV,2)
1920      H=0.07
1930      H02=0.5*H
1940      X0=0.5*(Y1(J)+Y2(J))
1950      Y0=0.25*(R1B(J)+R2B(J)+R1S(J)+R2S(J))
1960      Y00=-0.25*(R1B(J)+R2B(J)+R1S(J)+R2S(J))
1970      IF(J.LE.N) GO TO 2
1980      NF=NF+1

```

Figure F.1 (Sheet 4 of 5).

MOMENT
UTILITY PROGRAM FOR WEIGHT, CG, AND MOMENT-OF-INERTIA COMPUTATION

```
1990  N=N+NSECT(NF)
2000  2 FIG=NF
2010  IF(DX(J)*SFINV.LT.H) GO TO 3
2020  IF(.5*SFINV*(R1B(J)+R2B(J)-R1S(J)-R2S(J)).LT.H) GO TO 3
2030  CALL NUMBER(X0*SFINV-H02,Y0*SFINV-H02,H,FIG,0.,-1)
2040  CALL NUMBER(X0*SFINV-H02,Y00*SFINV-H02,H,FIG,0.,-1)
2050  3 CONTINUE
2060  CALL PLOT(0.,0.,999)
2070  RETURN
2080  END
```

Figure F.1 (Sheet 5 of 5).

```

001  NFIGS
002  NSECT(1),IDENS(1)
003  W(1) or DENSITY(1) {depending on IDENS(1)}
004  R2B(1),R2S(1),R1B(1),R1S(1),Y1(1),DX(1)
:
:
R2B(NSECT(1)),R2S(NSECT(1)),...,DX(NSECT(1))

050  NSECT(2),IDENS(2)
051  W(2) or DENSITY(2)
052  R2B(1+NSECT(1)),... ,DX(1+NSECT(1))
:
:
R2B(NSECT(2)+NSECT(1)),... ,DX(NSECT(2)+NSECT(1))

:
:
500  NSECT(NFIGS),IDENS(NFIGS)
501  W(NFIGS) or DENSITY(NFIGS)
502  R2B(1 +  $\sum_{i=1}^{NFIGS-1}$  NSECT(i)),... ,DX(1 +  $\sum_{i=1}^{NFIGS-1}$  NSECT(i))
:
:
550  R2B( $\sum_{i=1}^{NFIGS}$  NSECT(i)), R2S( $\sum_{i=1}^{NFIGS}$  NSECT(i)),...,DX( $\sum_{i=1}^{NFIGS}$  NSECT(i))

```

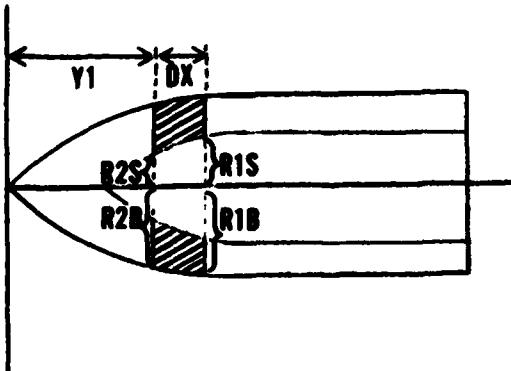


Figure F.2 Format of MOMENT's input file.

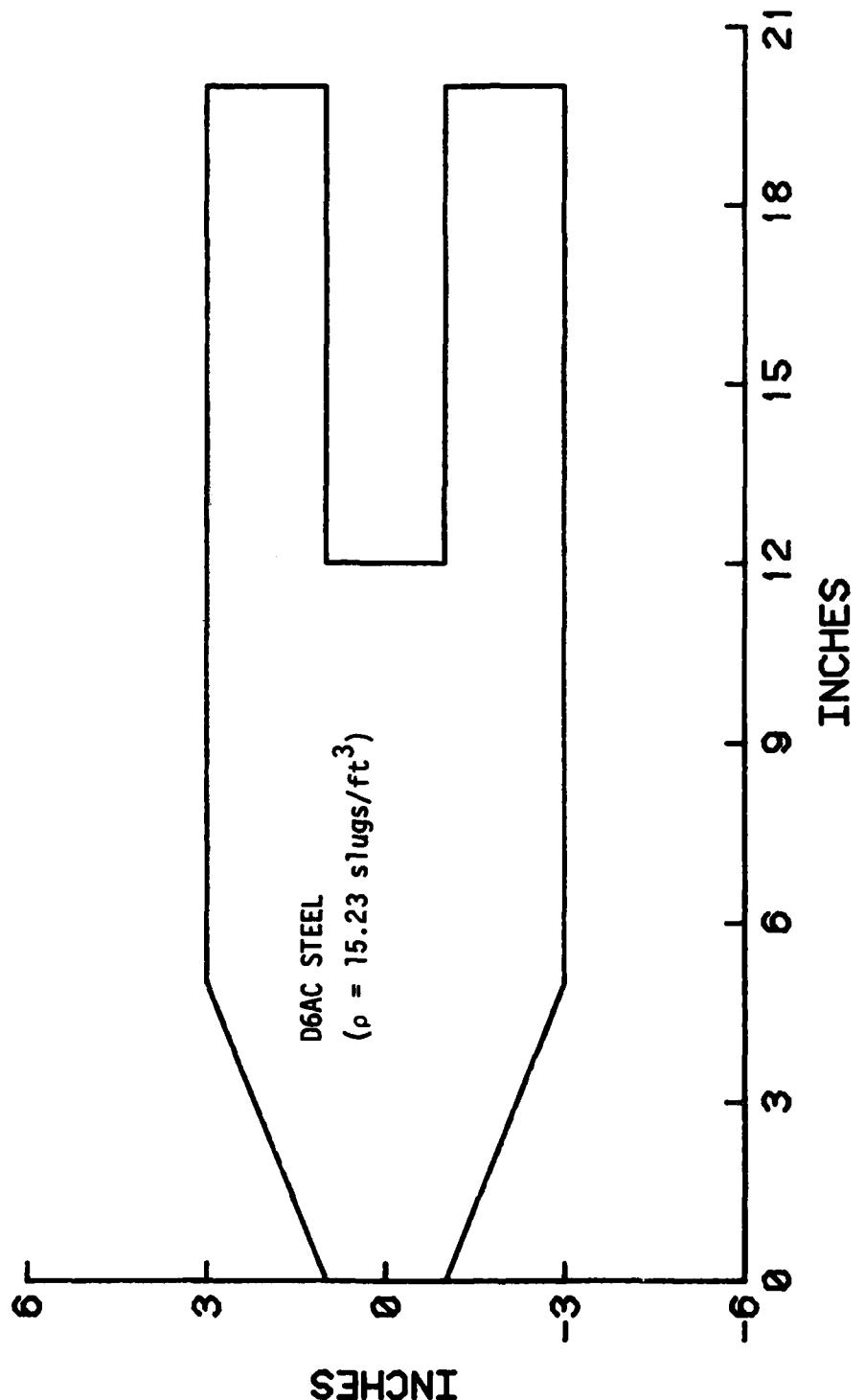


Figure F.3 Blunted conical-nosed projectile.

001 1
002 3,2
003 15.23
004 1,0,3,0,0,5
005 3,0,3,0,5,7
006 3,1,3,1,12,8

Figure F.4 Input file for blunted conical-nosed projectile.

\$OLD MOMENT

\$FRN

INPUT FILENAME

=BLNTMOM

DO YOU WANT TO CALCULATE MOMENTS?

=YES

TOTAL TRANSVERSE MOMENT OF INERTIA = 8.1801E-01 SLUG-FEET**2
= 3.7930E 03 LB.-INCHES**2
= 1.1090E 00 KG.-METERS**2

TOTAL POLAR MOMENT OF INERTIA = 1.2768E-01 SLUG-FEET**2
= 5.9201E 02 LB.-INCHES**2
= 1.7309E-01 KG.-METERS**2

TOTAL PROJECTILE WEIGHT = 1.3255E 02 LBS.
= 6.0124E 01 KG.

CENTER OF GRAVITY LOCATED 10.9664 INCHES FROM NOSE TIP
(27.8546 CM.)

DO YOU WANT TO PLOT?

=YES

Figure F.5 Time-sharing input-output sequence for blunted conical-nosed projectile.

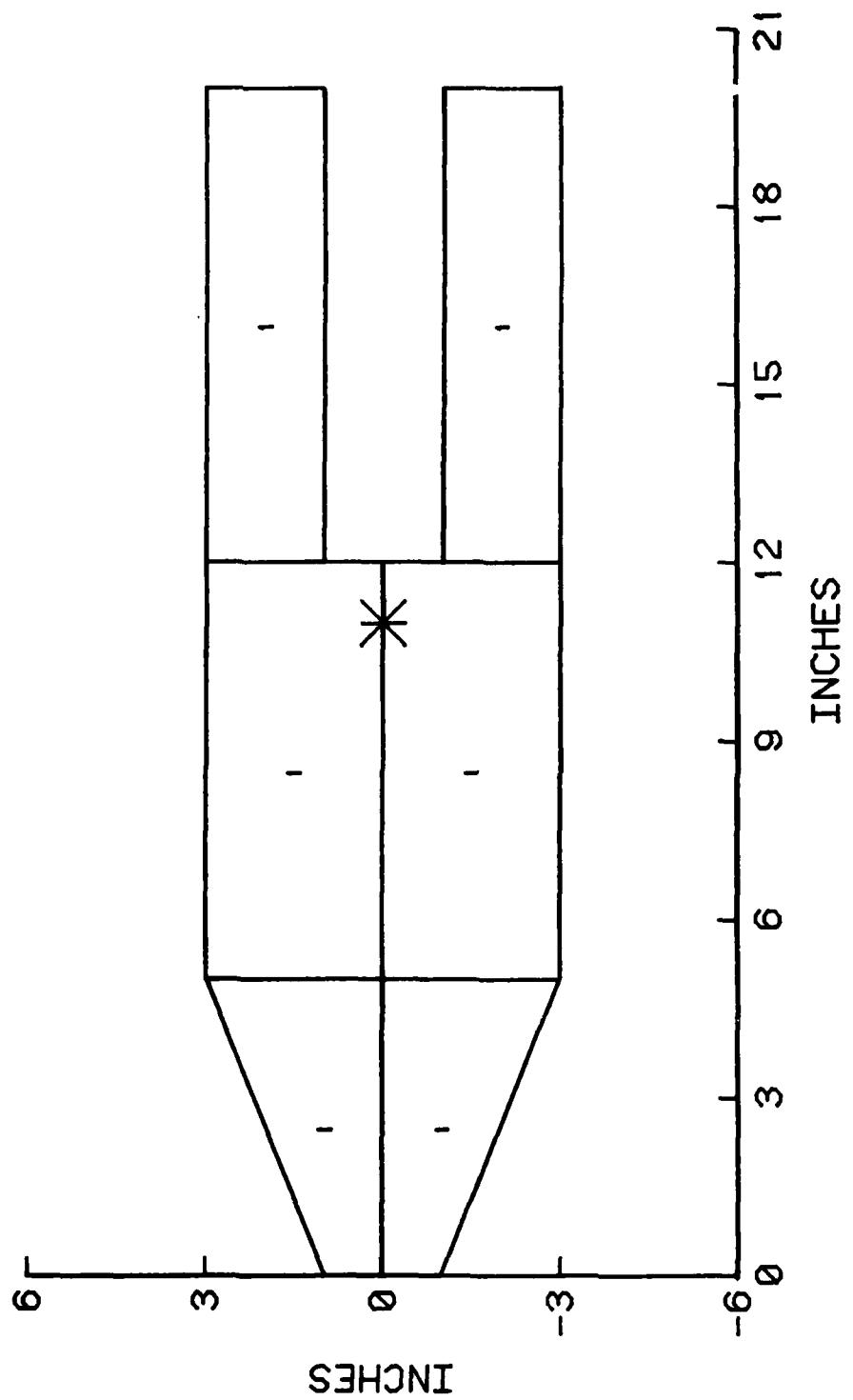
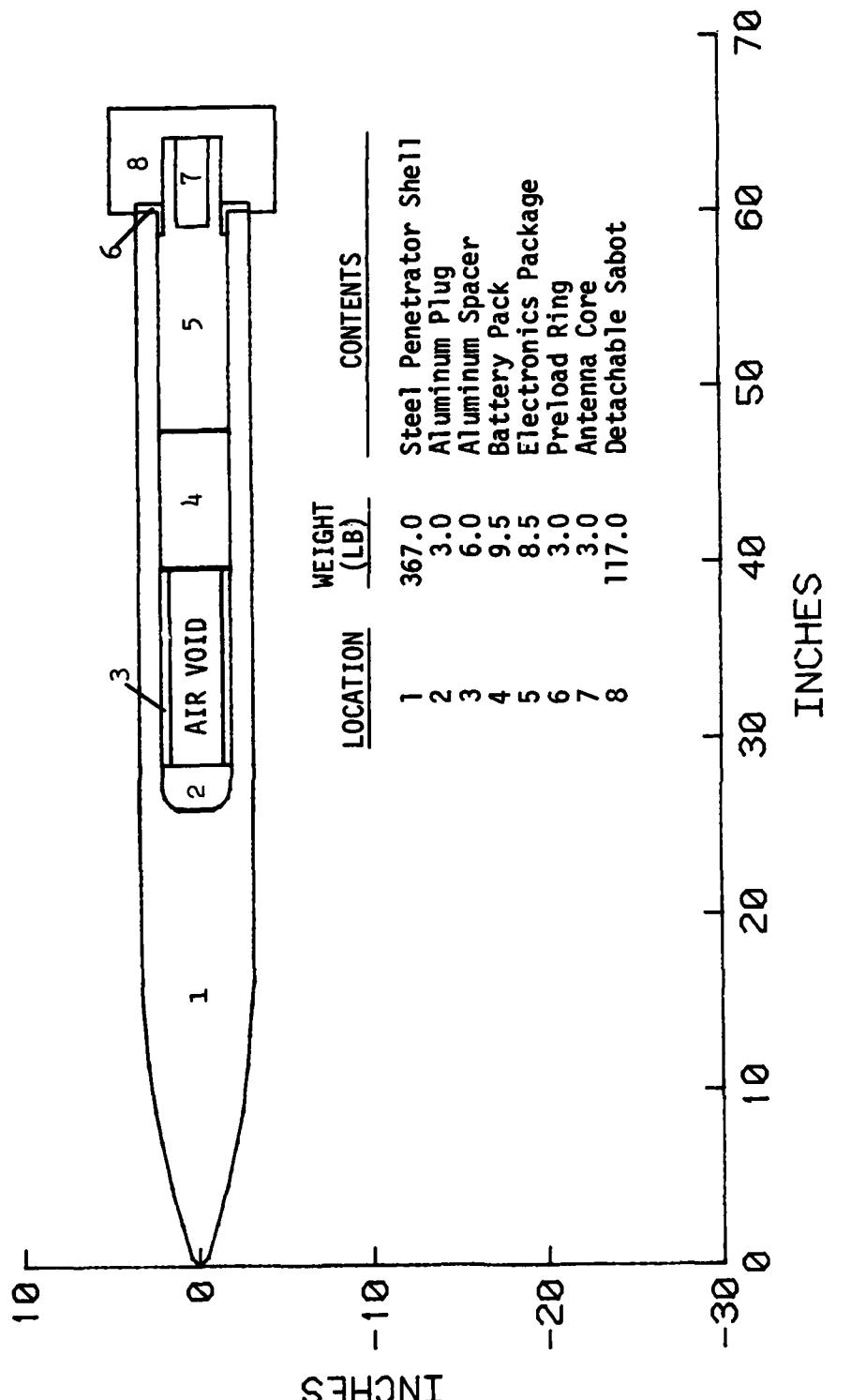


Figure F.6 MOMENT-generated plot for blunted conical-nosed projectile.



F13

Figure F.7 6.5-inch penetrator used in DNA/Watching Hill test in July 1974.

001 8
002 15,1
003 367.
004 0,0,.47,0,0,.44
005 .47,0,1.04,0,.44,1.92
006 1.04,0,1.6,0,2.36,2.14
007 1.60,0,2.16,0,4.5,2.58
008 2.16,0,2.54,0,7.08,1.76
009 2.54,0,2.85,0,8.84,2.6
010 2.85,0,3.02,0,11.44,1.72
011 3.02,0,3.16,0,13.16,1.76
012 3.16,0,3.22,0,14.92,1.32
013 3.22,0,3.25,0,16.24,9.74
014 3.25,.95,3.25,1.44,25.98,.2
015 3.25,1.44,3.25,1.84,26.18,.5
016 3.25,1.84,3.25,1.97,26.68,.52
017 3.25,1.97,3.25,2.,27.2,1.3
018 3.25,2,3.25,2.,28.5,31.5
019 4,1
020 3.
021 .95,0,1.44,0,25.98,.2
022 1.44,0,1.84,0,26.18,.5
023 1.84,0,1.97,0,26.68,.52
024 1.97,0,2,0,27.2,1.3
025 1,1
026 6.
027 2,1.48,2.,1.48,28.5,11.09
028 1,1
029 9.5
030 2,0,2,0,39.59,7.8
031 4,1
032 8.5
033 2,0,2,0,47.39,11.45
034 1.63,0,1.63,0,58.84,.4
035 1.63,.85,1.63,.97,59.24,.4
036 1.63,.97,1.63,.97,59.64,4.6
037 2,1
038 3.
039 2,1.63,2,1.63,58.84,1.16
040 3.25,1.63,3.25,1.63,60.,.5
041 2,1
042 3.
043 .85,0.,.97,0.,.59.24,.4
044 .97,0.,.97,0.,.59.64,4.6
045 3,1
046 117.
047 4.75,3.25,4.75,3.25,60.,.5
048 4.75,1.63,4.75,1.63,60.5,3.74
049 4.75,0.,4.75,0.,64.24,1.76

Figure F.8 Input file for Watchung Hill penetrator.

*OLD MOMENT

*FRN

INPUT FILENAME

=SABOT

DO YOU WANT TO CALCULATE MOMENTS?

=YES

TOTAL TRANSVERSE MOMENT OF INERTIA = 4.1062E 01 SLUG-FEET**2
= 1.9040E 05 LB.-INCHES**2
= 5.5667E 01 KG.-METERS**2

TOTAL POLAR MOMENT OF INERTIA = 8.0501E-01 SLUG-FEET**2
= 3.7327E 03 LB.-INCHES**2
= 1.0913E 00 KG.-METERS**2

TOTAL PROJECTILE WEIGHT = 5.1700E 02 LBS.
= 2.3451E 02 KG.

CENTER OF GRAVITY LOCATED 39.1027 INCHES FROM NOSE TIP
(99.3208 CM.)

DO YOU WANT TO PLOT?

=YES

Figure F.9 Time-sharing input-output sequence
for Watchung Hill penetrator.

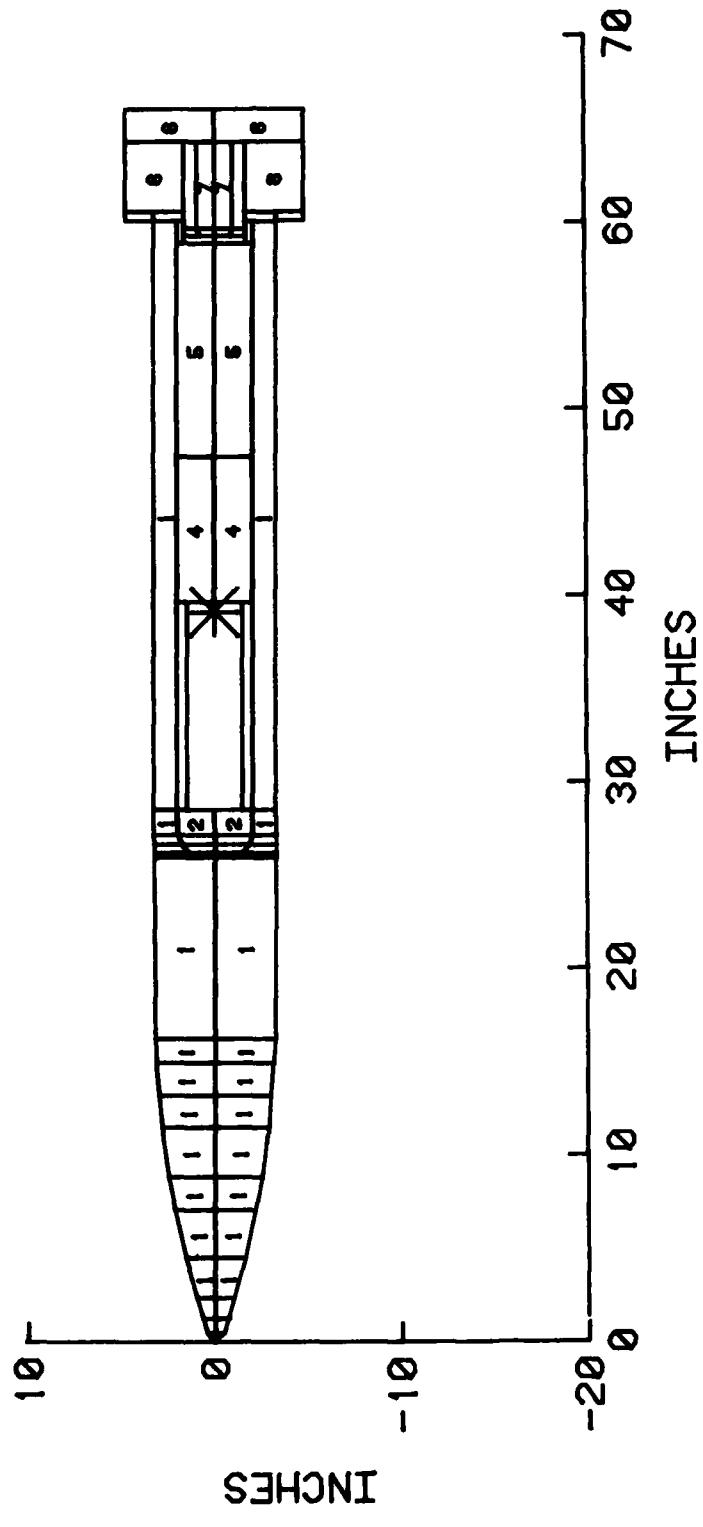


Figure F.10 MOMENT-generated plot for Watching Hill penetrator.

APPENDIX G
LISTING OF PROGRAM PENCO2D

PENCO2D
WES' TWO-DIMENSIONAL PENETRATION CODE

CPENCO2D

```

PARAMETER NMAX=7
DIMENSION RR0(70),SS0(70),RR1(70),SS1(70),QDECEL(200),QSPEED(200)
DIMENSION JQSET(20),VAVE(3),VPR(3),IQ(50),QPATH(200),QGMP(200)
COMMON/TRANS/SN0,XCG0/BACK/ALPHA/INCR/DGAMAD,DALFAD,NUMSTP
DIMENSION ZDUM(4),YDUM(4),QM(280),QN(280),Q1(200),Q2(200)
DIMENSION TIME(200),QLTACC(200),QAXACC(200),QCGMOM(200)
COMMON/TPLOT/TITLE,NLAY,NUMPLTS,ICH00Z(8),ZCUT/WT2/XIINV,XM
COMMON/LABELS/YLL,YLR,XL,XR,NCX,NCYL
CHARACTER X18YLL,XL,YLR*19,XR*19,TITLE*45
COMMON/PXTRA/STEST,PICON,ARATIO,NEL/X2/DELPREV
COMMON/PLTPR/FIRSTX,DELTAX,FIRSTY,DELTAY
COMMON/BIGS/ADA(70),CAT(70),SAT(70)/FIXT/SB,SC,SE,LOGCUT
COMMON/MSUN/ZSTOP,YSTOP,GAMSTP,VEL,ZSHIFT
COMMON/HOPE/VALUE(6),DER(6),THETAD(70),THETND,THETFD,
&TIMEF,TIMEI,GAMMAD,FREQI,NSTP,ALSTP/DOPE/W1I,FYP,NEMOV2,ALPHAD
COMMON/DARE/WEIGHT,XICG,AGRAB,FZP,VELF
COMMON/SHAP/DELS(70),S(70),R(70),THETA(70),THETAF,RB,EJ,
&RJ,RN,RC,RO,XFA,XFF,XLP,XCG,SN,THETAN
COMMON/FINK/ZM(NMAX),GAMI,RADDEG,DEGRAD,R90,R180,R270,R360,IOPSHP
COMMON/DAN4/XSALP(24),XCALP(24)/ROUND/ALAS,ELAS
COMMON/NEWT/FRAD,FANG,DTMAX,DTMIN,FREKOT,IREDUC,NUMNOS,NPRINT
COMMON/INTERM/AKASER(70,NMAX),BKBSER(70,NMAX),GKcate(NMAX),
&HRD1(NMAX),HRD2(NMAX)
COMMON/YOUNG/LMEDIA(NMAX),SNUM(NMAX),SHRGD(NMAX),DENSITY(NMAX),
&YIELD(NMAX)/SEPAR/SYNEFEE,TANFEE,PHIMIN,RCAV(70),S0(10)
LOGICAL LOGDT,LMEDIA,LPRNT,LOGCUT
REAL MU
DATA MU/1.633E4/,BET/0.1744/,GAM/90.28/
DATA AK/1.253/,BK/3.67/,CK/5.9/
FSQ(J,X)=SQRT(X(J)**2+X(J+1)**2)
AGRAB=386.07
R360=6.2831853; R180=0.5*R360; R90=R360*0.25; R270=R360*0.75
RADDEG=R360./R360
DEGRAD=R360/360.
SZE8=3.
CALL PLOTS(,,3)
CALL PLOT(0.,8.,-3)
LMEDIA(1)=T.
SNUM(1)=1.E10
ZM(1)=0.
1000 WRITE(6,4)
CALL READIN
FRADSV=FRAD
FANGSV=FANG
VCHKSV=.2*VEL
NUMB=0; SN0=SN; XCG0=XCG; IBACK=1
AKA=AK*MU
BKB=BK*SQRT(BET)
GKC=CK*GAM
PICON=R180/NEMOV2
IF(ELAS) 15,15,16
15 ELAS=0.01
16 ALAS=R90/ELAS
IMULT=NEMOV2/2
1001 GAMI=GAMMAD*DEGRAD

```

PENCO2D
WES' TWO-DIMENSIONAL PENETRATION CODE

```
GULF=(GAMMAD-ALPHAD)*DEGRAD
THETAN=THETND*DEGRAD
THETAF=THETFD*DEGRAD
IF(IOPSHP.NE.0) GO TO 103
INL=NEL
DO 107 I=1,NEL
THETA(I)=THETAD(I)*DEGRAD
107 CONTINUE
103 S(1)=XCG - DELS(1)/2.
NUMB=NUMB + 1
IF(NUMB.GT.NUMSTP) GO TO 1000
XCG=XCG0
SN=SNO
FRAD=FRADSV
ICNT=0
VCHEK=VCHKSV
FANG=FANGSV
YITO=XCG*SIN(GAMI)
ZITO=-XCG*COS(GAMI) - ZSHIFT
LOGCUT=.F.
DYITO=-VEL*SIN(GULF)
DZITO=-VEL*COS(GULF)
IF(NUMB.EQ.1) GO TO 104
IF(IBACK.EQ.1) GO TO 835
CALL BACKWARD
IBACK=1
GO TO 835
104 IF(IOPSHP.NE.0) GO TO 106
DO 105 I=2,NEL
IM1=I-1
S(I)=S(IM1)-((DELS(IM1)+DELS(I))/2.)
105 CONTINUE
GO TO 200
106 CONTINUE
STEST = XCG - XLP
DELS(2)=DELS(1)
CALL CONSET
GOTO(110,120,130,140),IOPSHP
110 CALL SHP1($1000)
GO TO 201
120 CALL SHP2
GO TO 201
140 CALL SHP4
GO TO 201
130 CALL SHP3($1000)
201 INL=NEL
NEL=NEL + 1
S(NEL)=S(INL)-DELS(INL)/2.-.0005
DELS(NEL)=.001
R(NEL)=(R(INL) + DELS(INL)*TAN(THETA(INL))/2.)/2.
THETA(NEL)=-ATAN(R(NEL)/.0005)
200 CONTINUE
CALL SEPARATE
WRITE(6,2) IOPSHP,NEL
2 FORMAT("1 SHAPE IS",I2,4X,"NEL=",I5,/,4X,"I",
&9X,"R(I)",9X,"S(I)",7X,"THETA(I)",6X,"DELS(I)",6X,"RCAV(I)")
```

PENCO2D
WES' TWO-DIMENSIONAL PENETRATION CODE

```

41 CONTINUE
RMAX=0.; NLM=4*NEL+1; NEX=NLM-1
DO210 I=1,NEL
J2=2*I
J1=J2 - 1
K1=NLM - J2
K2=NLM - J1
DOVE2=DELS(I)/2.
QM(J1)=S(I) + DOVE2
QM(J2)=S(I) - DOVE2
QM(K2)=QM(J1)
QM(K1)=QM(J2)
DR=DOVE2*TAN(THETA(I))
QN(J1)=R(I) - DR
QN(J2)=R(I) + DR
QN(K2)=-QN(J1)
QN(K1)=-QN(J2)
RMAX=AMAX1(RMAX,R(I))
X=THETA(I)*RADDEG
WRITE(6,5) I,R(I),S(I),X,DELS(I),RCAV(I)
55 CONTINUE
RR0(I)=R(I)
SS0(I)=S(I)
J=NEL-I+1
RR1(I)=R(J)
210 SS1(I)=-S(J)
VALP=QM(I) - QM(2*NEL)
VBET=2.*RMAX
IF(VALP.LT.VBET)GOTO212
CALL SCALIT(QM,SZE8,NEX,1)
QN(NEX+1)=QM(NEX+1)
QN(NEX+2)=QM(NEX+2)
GOTO213
212 CALL SCALIT(QN,SZE8,NEX,1)
QN(NEX+1)=QN(NEX+1)
QN(NEX+2)=QN(NEX+2)
213 CONTINUE
NCX=25
XL=18H      DIST FROM CG (
XR=3HIN); NCYL=21
YLL=18HX-SECTION RADIUS (
YLR=3HIN)
SZD=SZE8+.15
CALL SYMBOL(1.,SZD,.2,16HPROJECTILE SHAPE,0.,16)
CALL PLOTIT(NEX,QN,QM,0,SZE8)
QN(NEX+1)=0.
DL=PICON
DO 10 J=1,IMULT
ALP = (J-.5)*DL
XSALP(J) = SIN(ALP)
10 XCALP(J) = COS(ALP)
DO 11 J=IMULT+1,NEMOV2
JJ=NEMOV2+1-J
XSALP(J) = XSALP(JJ)
11 XCALP(J) = -XCALP(JJ)
DO 915 JJ=1,NLAY

```

PENCO2D
WES' TWO-DIMENSIONAL PENETRATION CODE

```

IF(.NOT.LMEDIA(JJ)) GO TO 918
DO 916 I=1,NEL
  AKASER(I,JJ)=AKA/(2.*R(I)*SNUM(JJ))
916  BKBSER(I,JJ)=BKB*SQRT(2.*R(I))/SNUM(JJ)
  GKcate(JJ)=GKC/SNUM(JJ)**2
  GO TO 915
918  HRD1(JJ)=6.3298*YIELD(JJ)
  HRD2(JJ)=4.74735*SQRT(DENSITY(JJ)*YIELD(JJ))
915  CONTINUE
  DO 829 I=1,NEL
    ATEH = R90 - THETA(I)
    CAT(I)=COS(ATEH)
    SAT(I)=SIN(ATEH)
829  ADA(I)=DL*R(I)*DELS(I)/COS(THETA(I))
835  WRITE(6,4)
    VALUE(1)=YITO
    VALUE(2)=ZITO
    VALUE(3)=DYITO
    VALUE(4)=DZITO
    VALUE(5)=GAMI
    VALUE(6)=W1I
    VB=VEL
    W=WEIGHT; IQSET=1
    JQSET(1)=1
    JQSET(2)=NSTP+1
    CALL CALF
    ABALF=ABS(ALPHA)
    PLAST=0.
    MXZ=0
    IF(SN.LE.0.) SN=0.5*XLP
    SEGLEN=SN/NUMNOS
    IBACK=1
    NCNT1=0; TIND=TIMEI; LOGDT=.F.
    NF=1; FNF=FREKOT
    LPRNT=.F.
    IF(NPRINT.EQ.2) LPRNT=.T.
    DEL=SEGLEN/VB
    PATH=0.; NBODY=1
    PTST=NBODY*FREQI*XLP
    PTdif=PTST; IQ(1)=1
    JQ=1
    CALL FORCES(VALUE,TIMEI)
    WRITE(6,3) JQ,YITO,ZITO,GAMMAD,NCNT1,DYITO,DZITO,W1I,
    &MXZ,TIMEI,FYP,FZP,SB,DEL,FYP/W,FZP/W,SB/W,ALPHA,SC,
    &SC/W,SE,W
    Q1(1)=YITO; Q2(1)=ZITO; QGMP(1)=GAMI-R180; TIME(1)=TIMEI
    QLTACC(1)=SB/W; QAXACC(1)=SC/W; QCGMOM(1)=SE
    QDECEL(1)=-QAXACC(1); QSPEED(1)=VB/12.; QPATH(1)=0.
    QLTMIN=QLTACC(1); QLTMAX=QLTACC(1); QDECMX=QDECEL(1)
    DO 400 JQ=2,NSTP
220  CONTINUE
    IF(LOGDT) GO TO 227
    IF(PATH.GT.SN) GO TO 502
    DT1=SEGLEN/VB
    GO TO 504
502  DT1=FRAD*RO/VB

```

PENCO2D
WES' TWO-DIMENSIONAL PENETRATION CODE

```

504 ABV6=ABS(VALUE(6))
IF(ABV6.GT.0.5) GO TO 501
DEL=DT1
GO TO 503
501 DT2=FANG/ABV6
DEL=MIN(DT1,DT2)
503 DEL=MIN(DEL,DTMAX)
DEL=MAX(DEL,DTMIN)
227 CALL FORCES(VALUE,TIND)
VB=FSQ(3,VALUE)
IF(VB.LE.VELF) MXX=7
IF(TIND.GE.TIMEF) MXX=3
CALL CALF
ABALF=ABS(ALPHA)
IF(ABALF.GE.ALSTP) MXX=9
IF(ABALF.LT.90.) GO TO 224
CALL BACKWARD
IBACK=-IBACK
IQSET=IQSET+1
JQSET(IQSET)=JQ
GO TO 223
224 IF(MXX.GT.2) GO TO 223
IF(LPRNT) GO TO 20
IF(TIND.LT.FNF) GO TO 45
GO TO 22
20 IF(PATH.LT.FNF) GO TO 45
22 NF=NF+1
FNF=NF*FREKOT
GO TO 223
45 IF(LOGCUT) GO TO 288
IF(VALUE(2).GT.ZCUT) GO TO 288
IF(ABS(VALUE(6)).GT.0.05.OR.ABALF.GT.0.01) GO TO 288
IF(ABS(SB/W).GT.1.5) GO TO 288
LOGCUT=.T.
VALUE(3)=-VB*SIN(VALUE(5))
VALUE(4)= VB*COS(VALUE(5))
VALUE(6)=0.
GO TO 227
288 IF(LPRNT) GO TO 24
IF(TIND+DEL.GT.FNF) DEL=FNF-TIND
24 VPR(1)=VALUE(3)
VPR(2)=VALUE(4)
VPR(3)=VALUE(6)
186 DER(3)=DEL*FYP*XM
DER(4)=DEL*FZP*XM
IF(LOGDT) GO TO 187
IF(VB.GT.VELF+2.0*FSQ(3,DER)) GO TO 187
LOGDT=.T.
DEL=DELPREV
DELSAV=DEL
GO TO 186
187 IF(LOGCUT) GO TO 48
DER(6)=DEL*SEX*XIINV
GO TO 47
48 DER(6)=0.
47 DO 239 II=1,3

```

PENC02D
WES' TWO-DIMENSIONAL PENETRATION CODE

```

II2=II+2
IP=II
IF(II.NE.3) GO TO 238
II2=6
IP=5
238 VALUE(II2)=VALUE(II2) + DER(II2)
VAVE(II)=(VALUE(II2) + VPR(II))/2.
DER(IP)=VAVE(II)*DEL
239 VALUE(IP)=VALUE(IP) + DER(IP)
TIND=TIND + DEL
NCNT1=NCNT1+1
DELPREV=DEL
PATH=PATH + FSQ(1,DER)
IF(PATH.LT.PTST) GO TO 220
NBODY=NBODY + 1
PLAST=PATH
PTST=NBODY*FREQI*XLP
IQ(NBODY)=JQ
GO TO 220
223 IF(JQ.EQ.NSTP) MXX=2
IF(ABS(VALUE(1)).GT.YSTOP) MXX=4
IF(VALUE(2).LT.ZSTOP) MXX=5
IF(VALUE(2).GT.MAX(XCG,XLP-XCG)) MXX=6
V5=VALUE(5)*RADDEG
IF(V5.GT.GAMSTP) MXX=8
WRITE(6,3) JQ,VALUE(1),VALUE(2),V5,NCNT1,VALUE(3),VALUE(4),
& VALUE(6),MXX,TIND,FYP,FZP,SB,DELPREV,FYP/W,FZP/W,
& SB/W,ALPHA,SC,SC/W,SE,W
3 FORMAT(/,1X,6HSTEP =,I4,11X,13HY-POSITION =,E13.6,
& 4X,12HZ-POSITION =,E13.6,4X,14HPITCH ANGLE =,E13.6,
& 4X,9HCOUNT =,I4,/,22X,13HY-VELOCITY =,E13.6,4X,
& 12HZ-VELOCITY =,E13.6,4X,14HPITCH RATE =,E13.6,4X,
& 9HRETURN =,I4,/,1X,6HTIME =,E11.4,4X,13HY-FORCE =,
& E13.6,4X,12HZ-FORCE =,E13.6,4X,14HPITCH FORCE =,E13.6,
& 4X,7HDEL =,E11.4,/,22X,5HY-G'S,7X,1H =,E13.6,4X,5HZ-G'S,
& 6X,1H =,E13.6,4X,9HPITCH G'S,4X,1H =,E13.6,4X,7HALPHA =,
& E11.4,/,22X,13HAXIAL FORCE =,E13.6,4X,12HAXIAL G'S =,
& E13.6,4X,14HPITCH MOMENT =,1PE11.4,2X,
& 12HPROJ WEIGHT =,E11.4)
IF(LOGCUT) GO TO 19
IF(VB.GT.VCHEK) GO TO 19
IF(ICNT.GE.IREDUC) GO TO 19
VCHEK=0.5*VCHEK
FRAD=FRAD/3.
FANG=FANG/3.
ICNT=ICNT+1
19 CONTINUE
IF(LOGDT) DEL=DELSAV
Q1(JQ)=VALUE(1)
Q2(JQ)=VALUE(2)
QGMP(JQ)=VALUE(5)-R180
TIME(JQ)=TIND
QLTACC(JQ)=SB/W
QAXACC(JQ)=SC/W
QCGMOM(JQ)=SE
QDECEL(JQ)=-QAXACC(JQ)

```

PENC02D
WES' TWO-DIMENSIONAL PENETRATION CODE

```

QSPEED(JQ)=VB/12.
QPATH(JQ)=PATH
NCNT1=0
QLTMAX=MAX(QLTMAX,QLTACC(JQ))
QLTMIN=MIN(QLTMIN,QLTACC(JQ))
QDECMX=MAX(QDECMX,QDECCEL(JQ))
IF(MXX.GT.2) GO TO 450
400 CONTINUE
450 NBODY=NBODY + 1
IQ(NBODY)=JQ
IF(NBODY.EQ.2.OR.PATH-PLAST.GT.0.55*PTDIF) GO TO 255
NBODY=NBODY - 1
IQ(NBODY)=JQ
255 JQSET(IQSET+1)=JQ+3
IF(JQ.LT.4) GO TO 777
WRITE(6,420) QPATH(JQ),QPATH(JQ)/12.,(TIME(JQ)-TIMEI)*1000.,
&QSPEED(JQ),QLTMIN,QLTMAX,QDECMX
420 FORMAT(//3X,"FINAL STATISTICS",/3X,16(1H*),/1X,"PATH LENGTH ",
&"TRAVELED=",E13.5," IN. (",E13.5," FT.)",/1X,"TIME ELAPSED=",E13.5," MSEC",/1X,"FINAL SPEED=",E13.5," FPS",/1X,"MIN. LATERAL ",E13.5," ACCELERATION=",E13.5," G'S",/1X,"MAX. LATERAL ACCELERATION=",E13.5," G'S")
&E13.5," G'S",/1X,"MAX. AXIAL DECELERATION=",E13.5," G'S")
TADD=AMAX1(XLP,2*R0)
CALL SCALIT(Q1,SZE8,JQ,1)
X1=Q1(JQ+1); DX1=Q1(JQ+2)
CALL SCALIT(Q2,SZE8,JQ,1)
X2=Q2(JQ+1); DX2=Q2(JQ+2)
63 CONTINUE
ZDUM(1)=Q2(1); ZDUM(2)=Q2(1); YDUM(1)=Q1(1); YDUM(2)=Q1(1)
DO 234 J=1,JQ
ZDUM(1)=AMIN1(Q2(J)-TADD,ZDUM(1))
ZDUM(2)=AMAX1(Q2(J)+TADD,ZDUM(2))
YDUM(1)=AMIN1(Q1(J)-TADD,YDUM(1))
234 YDUM(2)=AMAX1(Q1(J)+TADD,YDUM(2))
CALL SCALIT(ZDUM,SZE8,2,1)
CALL SCALIT(YDUM,SZE8,2,1)
X1=YDUM(3)
DX1=YDUM(4)
X2=ZDUM(3)
DX2=ZDUM(4)
Q2(JQ+1) = X2
Q2(JQ+2) = DX2
Q1(JQ+1) = X1
Q1(JQ+2) = DX1
DELSCAL = AMAX1(DX1,DX2)
IF(DELSCAL-DX1) 350,352,350
352 Q2(JQ+2) = DELSCAL
CALL SETUP(X2,SZE8,DELSCAL,ISTRRT,DX2,DX1)
Q2(JQ+1) = ISTRRT * DELSCAL
GO TO 354
350 Q1(JQ+2) = DELSCAL
CALL SETUP(X1,SZE8,DELSCAL,ISTRRT,DX1,DX2)
Q1(JQ+1) = ISTRRT * DELSCAL
354 YLL=18H
XL=18H
Z(
Y(
YLR=3HIN); NCYL=36

```

PENC02D
WES' TWO-DIMENSIONAL PENETRATION CODE

```

XR=YLR; NCX=NCYL
FIRSTX = Q1(JQ+1)
DELTAX = Q1(JQ+2)
FIRSTY = Q2(JQ+1)
DELTAY = Q2(JQ+2)
DO 333 J=1,NBODY
JUSE=IQ(J)
INDEX=2
330 IF(JUSE.LT.JQSET(INDEX)) GO TO 331
INDEX=INDEX + 1
GO TO 330
331 INDEX=INDEX - 1
INDY=(INDEX/2)*2
IF(INDY.EQ.INDEX) GO TO 447
CALL PLOTPRO(Q1(JUSE),Q2(JUSE),QGMP(JUSE),SS0,RR0)
GO TO 333
447 CALL PLOTPRO(Q1(JUSE),Q2(JUSE),QGMP(JUSE),SS1,RR1)
CONTINUE
NCHS=-FIRSTY/DELTAY
XNC=NCHS
CALL AXIS13(0.,XNC,1H ,1,0.,SZE8,-1,0,0.,0.,1.,1,0)
XST=-.5
IZE=SZE8
DO 125 LZ=1,IZE
XST=XST+1.0
CALL CALCMR(XST,XNC,0,1)
125 CALL CALCMR(XST,XNC-.07,16,1)
JO=1
123 JO=JO+1
IF(JO.GT.NMAX) GO TO 124
IF(ZM(JO)+ZDUM(1)) ,124,124
YST=ZM(JO)/DELTAY
YST=XNC-YST
CALL CALCMR(0.,YST,0,1)
CALL CALCMR(SZE8,YST,15,1)
GO TO 123
124 CALL PLOTIT(JQ,Q2,Q1,0,SZE8)
IF(NUMPLTS.EQ.0) GO TO 1000
XL=18H TIME (
XR=4HSEC); NCX=34
CALL SCALIT(TIME,SZE8,JQ,1)
DO 149 J=1,NUMPLTS
GO TO (131,132,133,134,135,136,137,138),ICH00Z(J)
CXXXXXX PLOT AXIAL ACCELERATION VS TIME XXXXXX
131 YLL=18H AXIAL ACC (
YLR=4HG'S); NCYL=29
CALL PLOTIT(JQ,QAXACC,TIME,1,SZE8)
GO TO 149
CXXXXXX PLOT AXIAL DECELERATION VS TIME XXXXXX
132 YLL=18H AXIAL DEC (
YLR=4HG'S); NCYL=29
CALL PLOTIT(JQ,QDECEL,TIME,1,SZE8)
GO TO 149
CXXXXXX PLOT LATERAL ACCELERATION VS TIME XXXXXX
133 YLL=18H LATERAL ACC (
YLR=4HG'S); NCYL=27

```

PENCO2D
WES' TWO-DIMENSIONAL PENETRATION CODE

```

CALL PLOTIT(JQ,QLTACC,TIME,1,SZE8)
GO TO 149
C***** PLOT CG MOMENT VS TIME *****
134 YLL=18H      CG MOMENT (
YLRL=6HLB-IN); NCYL=31
CALL PLOTIT(JQ,QCGMOM,TIME,1,SZE8)
GO TO 149
C***** PLOT VELOCITY VS TIME *****
135 YLL=18H      VELOCITY (
YLRL=4HFPS); NCYL=30
CALL PLOTIT(JQ,QSPEED,TIME,1,SZE8)
GO TO 149
C***** PLOT Y-DISPLACEMENT VS TIME *****
136 YLL=18H      Y (
YLRL=3HIN); NCYL=36
CALL PLOTIT(JQ,Q1,TIME,1,SZE8)
GO TO 149
C***** PLOT Z-DISPLACEMENT VS TIME *****
137 YLL=18H      Z (
YLRL=3HIN); NCYL=36
CALL PLOTIT(JQ,Q2,TIME,1,SZE8)
GO TO 149
C***** PLOT AXIAL DECELERATION VS PATH LENGTH *****
138 YLL=18H      AXIAL DEC (
YLRL=4HG'S); NCYL=29
XL=18H      PATH LENGTH (
XR=3HIN); NCX=26
CALL SCALIT(QPATH,SZE8,JQ,1)
CALL PLOTIT(JQ,QDECEL,QPATH,1,SZE8)
149 CONTINUE
777 CONTINUE
IF(DALFAD) 1003,1002,1003
1002 IF(DGAMAD) 1003,1000,1003
1003 GAMMAD=GAMMAD + DGAMAD
ALPHAD=ALPHAD + DALFAD
GO TO 1001
4 FORMAT(1H1)
5 FORMAT(2X,I3,5X,F10.5,4F13.5)
END

```

```

C***** CALCULATE CAVITY-RADIUS ARRAY AND SEPARATION POINT(S) *****
SUBROUTINE SEPARATE
COMMON/SEPARATE/SYNEFEE,TANFEE,PH,RCAV(70),S0(10)
&/PXTRA/DUM(3),NEL/SHAP/DELS(70),
&S(70),R(70),TH(70),THETAF,DUMN(5),R0,XFA,XFF,XLP,XCG,SN
RCAV(1)=R(1)
J=2
I=0
10 IF(PH) 15,15,11
11 IF(PH.GT.TH(J)) GO TO 20
15 RCAV(J)=MAX(R(J),RCAV(J-1))
J=J+1
IF(J.GT.NEL) GO TO 100
GO TO 10
20 I=I+1

```

PENCO2D
WES' TWO-DIMENSIONAL PENETRATION CODE

```
S0(I)=S(J)+0.5*DELS(J)
R0=R(J)-0.5*DELS(J)*TAN(TH(J))
ROSQ=R0**2
RTF=2*R0*TANFEE
22 RCAV(J)=SQRT(ROSQ+RTF*(S0(I)-S(J)))
IF(RCAV(J).LT.R(J)) GO TO 15
J=J+1
IF(J.GT.NEL) GO TO 100
GO TO 22
100 I=I+1
S0(I)=-1.E20
RETURN
END
```

```
***** DETERMINE AXES STARTING LOCATION FOR TRAJECTORY PLOT *****
SUBROUTINE SETUP(X,SIZE,DSCALE,ISTRRT,D1,D2)
ISTRRT = X/SIZE
ISTRRT = ISTRRT - 4
15 IF(ISTRRT*DSCALE-X) 20,20,10
20 ISTRRT = ISTRRT + 1
GO TO 15
10 ISTRRT = ISTRRT - 1
IF(D1.LT.(.75*D2)) ISTRRT = ISTRRT - 1
RETURN
END
```

```
***** PLOT PROJECTILE SHAPE AT (YCG,ZCG) ON TRAJ. PLOT *****
SUBROUTINE PLTTPR(YCG,ZCG,GAMMAP,S,R)
COMMON/PLTPR/FIRSTX,DELTAX,FIRSTY,DELTAY
COMMON/PXTRA/DUMMY(3),NEL
DIMENSION S(1),R(1),X(150),Y(150)
SGMP=SIN(GAMMAP)
CGMP=COS(GAMMAP)
DO 10 J=1,NEL
ONE=S(J)*SGMP
TWO=S(J)*CGMP
THREE=R(J)*SGMP
FOUR=R(J)*CGMP
X(J)=ONE + FOUR + YCG
Y(J) = ZCG - TWO + THREE
JJ = 2*NEL - J + 1
Y(JJ)=ONE - FOUR + YCG
10 Y(JJ)=ZCG - (TWO + THREE)
NPTS = 2*NEL + 1
X(NPTS) = X(1)
Y(NPTS) = Y(1)
X(NPTS+1) = FIRSTX
X(NPTS+2) = DELTAX
Y(NPTS+1) = FIRSTY
Y(NPTS+2) = DELTAY
CALL LINE(X,Y,NPTS,1,0,12)
CALL SYMBOL((YCG-FIRSTX)/DELTAX,(ZCG-FIRSTY)/DELTAY,.07,11
```

PENCO2D
WES' TWO-DIMENSIONAL PENETRATION CODE

2,0.,-1)
RETURN
END

***** DRAW AND LABEL AXIS *****
SUBROUTINE AXIS13(X,Y,IBCD,NC,H,SIZE,NN,IXY,XMIN,DX,
& SPACE,ITIC,NP10)
COMMON/PAXIS/IBEGIN
DIMENSION IBCD(1)
DATA IBCD1/1H /
NCHARS=ABS(NC)
XY=IXY; ANGLE=XY*90.; TIC=ITIC
SPAC=SPACE; G=H; XT=X; YT=Y
GVERT=G; LINE=2
SIZ=SIZE; P10=NP10
IF(SIZ) 1,16,100
1 LINE=3
SIZ=-SIZ
100 ND=NN
NA=NN
IF(ND) 3,2,2
2 NDIG=ND + 1
GO TO 30
3 ND=0
NDIG=0
30 NSPACE=SIZ/SPACE + .5
FNSPAC=NSPACE
TL=FNSPAC*SPAC
IF(YT + TL*XY - 29.) 4,4,31
31 CALL SYMBOL(XT,YT+.1,.14,13HAXIS TOO LONG,90.,13)
RETURN
4 H02=.5*G
H07=H02/3.5
TEMP=-NP10
POWER=10.**TEMP
DELX=POWER*DX; XNUMB=POWER*XMIN
ANG=1. - XY
ALAB=1.
IF(NC) 5,6,6
5 ALAB=-1.
6 TICXY=(2.*XY-1.)*ALAB*TIC*.1
XTIC=TICXY*XY; YTIC=TICXY*ANG
TICM1=.16 - TIC*.05
LIN=3; NMAX=0; J=0
DO 12 I=J,NSPACE
NDIGIT=NDIG
CALL PLOT(XT,YT,LIN)
LIN=LINE
CALL PLOT(XT+XTIC,YT+YTIC,2)
IF(G) 7,11,7
7 TEMP=-ND
ITEMP=ALOG10(ABS(XNUMB)+.5*10.**TEMP) + 1.
IF(ITEMP) 70,70,71
70 ITEMP=1

PENC02D
WES' TWO-DIMENSIONAL PENETRATION CODE

```

71 NDIGIT=NDIGIT+ITEMP
    IF(ABS(XNUMB).LT.1.E-6) XNUMB=0.
    IF(XNUMB) 9,10,10
9   NDIGIT=NDIGIT+1
10  IF(NDIGIT-NMAX) 77,77,76
76  NMAX=NDIGIT
77  FDIGIT=NDIGIT
    CENTER=(7.*FDIGIT-3.)*H07*.5
    XANO=-ALAB*XY*(TICM1+CENTER) - CENTER
    YANO=ALAB*ANG*(TICM1+H02) - H02
    CALL NUMBER(XT+XANO,YT+YANO,G,XNUMB,0.,NA)
    XNUMB=XNUMB+DELX
11  CALL PLOT(XT,YT,3)
    XT=XT+SPAC*ANG
12  YT=YT+SPAC*XY
    IF(NCHARS.GT.1) GO TO 20
    IF(IBCD(1).EQ.IBCD1) RETURN
20  ANOLOC=IBEGIN*G*ALAB
    ANOCHR=(SIZ - NCHARS*GVERT)*.5
    XST=X - XY*ANOLOC + ANG*ANOCHR
    YST=Y + ANG*ANOLOC + XY*ANOCHR
    CALL SYMBOL(XST,YST,G,IBCD,ANGLE,NCHARS)
    IF(NP10.EQ.0) RETURN
    ANOLOC=(IBEGIN + ALAB*.5)*G*ALAB
    ANOCHR=(SIZ + GVERT*(40-NCHARS))*.5
    XST=X - XY*ANOLOC + ANG*ANOCHR
    YST=Y + ANG*ANOLOC + XY*ANOCHR
    CALL NUMBER(XST,YST,5.*H07,P10,ANGLE,-1)
16  RETURN
END

```

***** CONNECT PIECES OF AXIS LABEL TOGETHER *****

```

SUBROUTINE TLAB(KP,ICHOSE,TITLE,NC)
COMMON/LABELS/YLL,YLR,XL,XR,NCX,NCYL
CHARACTER *18YLL,XL,YLR*19,XR*19
CHARACTER LLAB*18,RLAB*19,TITLE*42
CHARACTER FMT1*17,FMT2*20,FMT3*17
DATA FMT1/17H(A18,"10  ",A19)//,FMT2/20H(A18,"10  ",A19," ")/,
&      FMT3/17H(A18,A19,"  ")/
    IF(ICHOSE.EQ.2) GO TO 11
    LLAB=XL; RLAB=XR; NC=NCX
    GO TO 13
11  LLAB=YLL; RLAB=YLR; NC=NCYL
13  IF(KP) 21,22,23
22  NSP=0
    ENCODE(TITLE,FMT3) LLAB,RLAB
    GO TO 25
23  IF(KP-9) 24,24,21
24  NSP=4
    ENCODE(TITLE,FMT2) LLAB,RLAB
    GO TO 25
21  NSP=5
    ENCODE(TITLE,FMT1) LLAB,RLAB
25  NC=NC+NSP

```

PENCO2D
WES' TWO-DIMENSIONAL PENETRATION CODE

RETURN
END

```
***** DETERMINE AXIS ANNOTATION STARTING LOCATION *****
SUBROUTINE ANOSTRT(LOCATE,KLOG,Q1,QSIZE,DELQ,RINC,NN)
POWER=10.**(-KLOG)
ANOMIN=Q1*POWER
ANOMAX=ANOMIN + QSIZE*DELQ*POWER/RINC
AMINAB=ABS(ANOMIN)
AMAXAB=ABS(ANOMAX)
MXLOG=0; MNLOG=0
IF(AMINAB.GT.(1.0)) MNLOG=ALOG10(AMINAB)
IF(AMAXAB.GT.(1.0)) MXLOG=ALOG10(AMAXAB)
LOCATE=5 + MAX(MXLOG,MNLOG) + NN
IF(MXLOG-MNLOG) 10,10,20
10 IF(ANOMIN.LT.(0.)) LOCATE=LOCATE + 1
GO TO 30
20 IF(ANOMAX.LT.(0.)) LOCATE=LOCATE + 1
30 RETURN
END
```

```
***** PLOT YARRAY VS XARRAY *****
SUBROUTINE PLOTIT(JQ,YARRAY,XARRAY,ISCALE,SZ)
COMMON/PAXIS/IBEGIN
DIMENSION XARRAY(1),YARRAY(1)
CHARACTER TITLE*42,AAA*8/1H /
IF(ISCALE.EQ.0) GO TO 30
CALL SCALIT(YARRAY,SZ,JQ,1)
30 X1=YARRAY(JQ+1)
XDEL=YARRAY(JQ+2)
KP=0
IF(XDEL.GT.0.) KP=ALOG10(XDEL)
IF(KP.LT.0) KP=KP-1
CALL TLAB(KP,2,TITLE,NC)
CALL ANOSTRT(LOCATE,KP,X1,NC,XDEL,1.,1)
IBEGIN=LOCATE
CALL AXIS13(0.,0.,TITLE,NC,.09,NC,1,1,X1,XDEL,1.,1,KP)
XSTART=-.5
DO 10 L=1,NC
XSTART=XSTART+1.
CALL CALCM(0.,XSTART,0,1)
10 CALL CALCM(0.07,XSTART,16,1)
X1=XARRAY(JQ+1)
XDEL=XARRAY(JQ+2)
KP=0
IF(XDEL.GT.0.) KP=ALOG10(XDEL)
IF(KP.LT.0) KP=KP-1
CALL TLAB(KP,1,TITLE,NC)
IBEGIN=5
CALL AXIS13(0.,0.,TITLE,-NC,.09,NC,1,0,X1,XDEL,1.,1,KP)
XSTART=-.5
DO 20 L=1,NC
```

PENC02D
WES' TWO-DIMENSIONAL PENETRATION CODE

```
XSTRT=XSTRT+1.  
CALL CALCMP(XSTRT,0.,0,1)  
20 CALL CALCMP(XSTRT,.07,16,1)  
CALL LINE(XARRAY,YARRAY,JQ,1,0,12)  
ENCODE(AAA,55) TITLE  
55 FORMAT(A8)  
IF(AAA.EQ.8H DIST) GO TO 25  
IF(AAA.EQ.8H DIST F) GO TO 25  
CALL HEADING  
25 CALL CALCMP(0.,0.,0000,2)  
RETURN  
END
```

```
***** DETERMINE SCALE FOR ARRAY GIVEN SZ AXIS LENGTH *****  
SUBROUTINE SCALIT(ARRAY,SZ,NPTS,INC)  
DIMENSION ARRAY(1),TIS(9),VM(9)  
DATA TIS/.07918,.17609,.30103,.39794,.47712,.60206,  
.6.69897,.77815,.90309/,VM/1.2,1.5,2.0,2.5,3.0,4.0,  
.5.0,6.0,8.0/  
TMX=0. ;TMN=0.  
DO 10 I=1,NPTS,INC  
TMX=AMAX1(TMX,ARRAY(I))  
TMN=AMIN1(TMN,ARRAY(I))  
10 CONTINUE  
SZUSE=SZ  
50 SUSE=(TMX-TMN)/SZUSE  
IF (SUSE) 12,118,12  
12 QLOG=0.  
IF (SUSE.GT.0.) QLOG= ALOG10(SUSE)  
KLOG=QLOG  
CHAR=KLOG  
TISSA=QLOG-CHAR  
VMULT=1.0  
IF(TISSA)5,110,20  
5 TISSA=TISSA+1.  
KLOG=KLOG-1  
IF(TISSA.LE.1.E-10) GO TO 110  
20 DO 41 J=1,9  
IF (TISSA.GT.TIS(J)) GO TO 41  
VMULT=VM(J)  
GO TO 110  
41 CONTINUE  
KLOG=KLOG+1  
110 ARRAY(NPTS+2)=VMULT*10.***KLOG  
XTEST=TMN/ARRAY(NPTS+2)  
NTEST=XTEST  
IF(NTEST.LE.XTEST) GO TO 120  
NTEST=NTEST-1  
GO TO 120  
118 ARRAY(NPTS+2)=0.  
NTEST=0  
120 ARRAY(NPTS+1)=NTEST*ARRAY(NPTS+2)  
RM_N=ARRAY(NPTS+1)  
RMAX=RMIN+SZ*ARRAY(NPTS+2)
```

PENCO2D
WES' TWO-DIMENSIONAL PENETRATION CODE

```
IF(TMX.GT.RMAX.OR.TMN.LT.RMIN)GO TO 60
RETURN
60  SZUSE=SZUSE*0.95
GO TO 50
END
```

***** PLOT DESCRIPTIVE HEADING ON FINAL PLOTS *****
SUBROUTINE HEADING
COMMON/SHAP/DMY(286),R0/TPLOT/TITLE
COMMON/MSUN/DUMN(3),VEL/DOPE/DUN(3),ALPHAD/DARE/WEIGHT
COMMON/HOPE/DUM2(86),GAMMAD
CHARACTER BADX2/2HD=/,BBDX8/8H IN., W=/
&BCDX8/8H LBS, V=/,BDDX4/4H FPS/,BEDX6/6HALPHA=/
&BFDX13/13H DEG., GAMMA=/,BGDX5/5H DEG./,TITLEX45
SZZ=6.
SZF=SZZ-.2
SZG=SZF-.2
CALL SYMBOL(.35,SZZ,.1,TITLE,0.,45)
R1=2.*R0
CALL SYMBOL(.35,SZF,.1,BAD,0.,2)
CALL NUMBER(999.,SZF,.1,R1,0.,2)
CALL SYMBOL(999.,SZF,.1,BBD,0.,8)
CALL NUMBER(999.,SZF,.1,WEIGHT,0.,2)
CALL SYMBOL(999.,SZF,.1,BCD,0.,8)
R1=VEL/12.
CALL NUMBER(999.,SZF,.1,R1,0.,-1)
CALL SYMBOL(999.,SZF,.1,BDD,0.,4)
CALL SYMBOL(.35,SZG,.1,BED,0.,6)
CALL NUMBER(999.,SZG,.1,ALPHAD,0.,2)
CALL SYMBOL(999.,SZG,.1,BFD,0.,13)
CALL NUMBER(999.,SZG,.1,GAMMAD,0.,-1)
CALL SYMBOL(999.,SZG,.1,BGD,0.,5)
RETURN
END

***** GEOMETRY GENERATOR FOR CONICAL NOSE *****
SUBROUTINE SHPI(*)
COMMON/SHAP/DELS(70),S(70),R(70),THETA(70),THETAF,RB,EJ,
&RJ,RN,RC,R0,XFA,XFF,XLP,XCG,SN,THETAN
THETAF=TAN(THETAF)
IF(THETAN-1.E-4) 1,1,2
1 IF(SN-1.E-4) 99,99,3
3 C3=R0/SM
THETAN=ATAN(C3)
GO TO 5
2 C3=TAN(THETAN)
SN=R0/C3
5 C1=XCG-SN
IFINISH=0
I=1
10 IF(I.NE.1) S(I)=S(I-1)-((DELS(I-1)+DELS(I))/2.)
IF(S(I).LT.C1)GOTO100

PENCO2D
WES' TWO-DIMENSIONAL PENETRATION CODE

```
R(I)=(XCG-S(I))xC3
THETA(I)=THETAN
GO TO 150
100 IF(S(I).LE.XFF.AND.S(I).GE.XFA) GO TO 120
R(I)=R0
THETA(I)=0.
GO TO 150
120 CONTINUE
R(I) = R0 + (XFF - S(I)) *TTHTEF
THETA(I)=THETAF
150 IF(IFINISH.EQ.1) RETURN
CALL CONVER(IFINISH,I)
GO TO 10
99 WRITE(6,98)
98 FORMAT(//,1X,"SHAPE 1 THETAN AND SN ARE ZERO.")
RETURN 1
END
```

***** GEOMETRY GENERATOR FOR CONICAL NOSE TOPPED BY A SPHERE *****

```
SUBROUTINE SHP2
COMMON/SHAP/DELS(70),S(70),R(70),THETA(70),THETAF,RB,EJ,
&RJ,RN,RC,R0,XFA,XFF,XLP,XCG,SN,THETAN/X2/DELPREV
TTHTEF=TAN(THETAF)
TTHETN=TAN(THETAN)
C1=XCG-SN
C4=XCG-RC
RNSQ=RN*RN
C=RN-RC
RB=SQRT(RNSQ-C*C)
IFINISH=0
I=1
SUMI=0.
10 IF(I.NE.1) S(I)=S(I-1)-((DELS(I-1)+DELS(I))/2.)
SUMIM1 = SUMI
SUMI=SUMI+DELS(I)
IF(S(I).LE.C4)GOT0120
IF(I.GT.1)GOT0110
XL=SQRT(RNSQ-(RN-DELS(I)))
THETA(I)=ATAN(XL/DELS(I))
C=RN-DELS(I)*.5
R(I)=SQRT(RNSQ-C*C)
GO TO 150
110 CONTINUE
C=RN-SUMIM1
RFI=SQRT(RNSQ-C*C)
C=C-DELS(I)*.5
R(I)=SQRT(RNSQ-C*C)
C=RN-SUMI
RAI=SQRT(RNSQ-C*C)
THETA(I)=ATAN((RAI-RFI)/DELS(I))
GO TO 150
120 CONTINUE
IF(S(I).LT.C1)GOT0130
THETA(I)=THETAN
```

PENCO2D
WES' TWO-DIMENSIONAL PENETRATION CODE

```

R(I)=RB+(C4-S(I))/TTHETN
GO TO 150
130 CONTINUE
IF(S(I).LE.XFF.AND.S(I).GE.XFA)GOTO145
R(I)=R0
THETA(I)=0.
GO TO 150
145 CONTINUE
R(I)=R0+ABS(S(I)-XFF)*TTHETF
THETA(I)=THETAF
150 IF(IFINISH.EQ.1) RETURN
II=I
CALL CONVER(IFINISH,I)
IF(I.EQ.II) SUMI=SUMI-DELPREV
GO TO 10
END

```

```

***** GEOMETRY GENERATOR FOR OGIVE WITH OPTIONAL CONE TIP (RB,SN>0) ****
SUBROUTINE SHP3(*)
COMMON/SHAP/DELS(70),S(70),R(70),THETA(70),THETAf,RB,EJ,
&RJ,RN,RC,R0,XFA,XFF,XLP,XCG,SN,THETAN
IF(RJ.LT.1.E-12) GO TO 110
PHI1=ARSIN(EJ/RJ)
T3=RJ*COS(PHI1)-R0
PHI2=ARCCOS(T3/RJ)
SN2=RJ*SIN(PHI2) - EJ
IF(RB.LT.1.E-12) SN=SN2
GO TO 130
110 IF(SN.LT.1.E-12) GO TO 120
SN2=SN
RB=0.
PHI3=ATAN(R0/SN)
T4=SN*.5/COS(PHI3)
T5=(EJ+SN*.5)/SIN(PHI3)
RJ=SQRT(T5*T5+T4*T4)
GO TO 130
120 CONTINUE
WRITE(6,1)
RETURN 1
1 FORMAT(1X,"SHAPE 3 RJ AND SN ARE ZERO")
130 CONTINUE
C1=XCG-SN
TTHETF=TAN(THETAf)
RJRJ = RJ * RJ
T8 = SN + EJ - XCG
IFINISH=0
I=1
IF(RB.LT.1.E-12) GO TO 14
PHIRB=ARCCOS((T3+RB)/RJ)
XX=SN-RJ*SIN(PHIRB)
TTHETN=RB/XX
THETAN=ATAN(TTHETN)
IF(DELS(1).GT.(0.8*XX)) GO TO 11
NCONE=IFIX(XX/DELS(1))+1

```

PENC02D
WES' TWO-DIMENSIONAL PENETRATION CODE

```

DXX=XX/NCONE
THETA(1)=THETAN
DELS(1)=DXX
XXX=DXX/2.
S(1)=XCG-XXX
R(1)=TTHETN*XXX
12 I=I+1
XXX=XXX+DXX
DELS(I)=DXX
IF(I.GT.NCONE) GO TO 10
S(I)=XCG-XXX
R(I)=TTHETN*XXX
THETA(I)=THETAN
GO TO 12
11 DELS(1)=XX
S(I)=XCG-XX/2.
THETA(I)=THETAN
R(I)=RB/2.
I=2
10 S(I)=S(I-1)-((DELS(I-1)+DELS(I))/2.)
14 IF(S(I).LE.C1) GO TO 140
T77=T8+S(I)
T7=SQRT(RJRJ-T77*T77)
R(I)=T7-T3
C=T77-DELS(I)*.5
RTH1=SQRT(RJRJ-C*C)
C=C+DELS(I)
RTH2=SQRT(RJRJ-C*C)
THETA(I)=ATAN((RTH1-RTH2)/DELS(I))
GO TO 150
140 IF(S(I).LE.XFF.AND.S(I).GE.XFA) GO TO 145
R(I)=R0
THETA(I)=0.
GO TO 150
145 CONTINUE
R(I)=R0+ABS(S(I)-XFF)*TTHETF
THETA(I)=THETAF
150 IF(IFINISH.EQ.1) RETURN
CALL CONVER(IFINISH,I)
GO TO 10
END

```

***** GEOMETRY GENERATOR FOR BLUNTED CONICAL NOSE *****

```

SUBROUTINE SHP4
COMMON/SHAP/DELS(70),S(70),R(70),THETA(70),THETAF,RB,EJ,
&RJ,RN,RC,RO,XFA,XFF,XLP,XCG,SN,THETAN
TTHETN=TAN(THETAN)
TTHETF=TAN(THETAF)
C XCG MUST BE GREATER THAN NOSE LENGTH
R(1)=0.5*RB
THETA(1)=ATAN(RB/.001)
DELS(1)=0.001
S(1)=XCG-.0005
RC= RB/TTHETN

```

PENC02D
WES' TWO-DIMENSIONAL PENETRATION CODE

```

SNT= R0/TTHETN
SN= SNT-RC
C1= XCG-SN
IFINISH=0
I=1
GO TO 150
10 S(I)=S(I-1)-((DELS(I-1)+DELS(I))/2.)
IF(S(I).LT.C1) GO TO 142
THETA(I)= THETAN
R(I)=(XCG-S(I))*TTHETN+RB
GO TO 150
142 CONTINUE
IF(S(I).LE.XFF.AND.S(I).GE.XFA) GO TO 144
R(I)= R0
THETA(I)=0.
GO TO 150
144 CONTINUE
R(I) = R0 + (XFF - S(I)) *TTHETF
THETA(I)=THETAF
150 IF(IFINISH.EQ.1) RETURN
CALL CONVER(IFINISH,I)
GO TO 10
END

```

```

C***** LONGITUDINAL ELEMENT THICKNESS GENERATOR (CONSTANT L/W FOR DA)
SUBROUTINE CONSET
COMMON/SHAP/DELS(70),S(70),R(70),THETA(70),THETAF,RB,EJ,
&RJ,RN,RC,RO,XFA,XFF,XLP,XCG,SN,THETAN
COMMON/PXTRA/STEST,PICON,ARATIO,NEL/X2/DELPREV
LOGICAL LOG,LOG2
FLDC(I,X)=S(I)-0.5*DELS(I)+X-XCG
ISET=0
SUZE=1.
SGNM=1.
LOG=.F.
LOG2=.F.
ISET=0
RETURN
ENTRY CONVER(IFINISH,I)
IF(LOG) GO TO 5
IF(I.EQ.1) GO TO 2
IF(ISET.EQ.1) GO TO 8
DELPREV=DELS(I)
DELS(I)=PICON*R(I)*COS(THETA(I))*ARATIO
TEST=ABS((DELPREV-DELS(I))/DELPREV)
IF(TEST.GT..003) RETURN
ISET=1
RETURN
8 IF(LOG2) GO TO 1
SUZEP=SUZE
SGNMP=SGNM
SUZE=FLDC(I,SN)
SGNM=SIGN(1.,SUZE)
IF(SGNMP-SGNM-1.) 1,5,3

```

PENCO2D
WES' TWO-DIMENSIONAL PENETRATION CODE

```

3 DELS(I)=FLOC(I-1,SN)
LOG=.T.
RETURN
5 LOG=.F.
LOG2=.T.
1 IF(FLOC(I,XLP).LE.0.) GO TO 100
2 I=I+1
ISET=0
IF(I.GT.70) GO TO 102
DELS(I)=DELS(I-1)
RETURN
100 SLENG=0.
DO 101 J=1,I-1
101 SLENG=SLENG+DELS(J)
DELS(I)=XLP-SLENG
NEL=I
IFINISH=1
RETURN
102 WRITE(6,103)
103 FORMAT(//,3X,"SUBROUTINE CONVER...I>70...DECREASE NEMOV2 OR ",
& "INCREASE ARATIO.")
STOP
END

```

```

***** CALCULATE COMPRESSIVE NORMAL STRESS, SUM FORCES & MOMENTS ***
SUBROUTINE FORCES(VALUE,TIND )
PARAMETER NMAX=7
COMMON/YOUNG/LMEDIA(NMAX),SNUM(NMAX),SHRGD(NMAX),DENSITY(NMAX),
&YIELD(NMAX)/INTERM/AKASER(70,NMAX),BKBSER(70,NMAX),GKcate(NMAX),
&HRD1(NMAX),HRD2(NMAX)
LOGICAL LMEDIA,LOGCUT
COMMON/PXTRA/DUMN(3),NEL/BACK/ALPHA
COMMON/FIXT/SB,SC,SE,LOGCUT
COMMON/SEPAR/E/SYNEFEE,TANFEE,PHIMIN,RCAV(70),S0(10)
COMMON/DAN4/XSALP(24),XCALP(24)/ROUND/ALAS,ELAS
COMMON/BIGS/ADA(70),CAT(70),SAT(70)
COMMON/DOPE/W1I,FYP,NEMOV2,ALPHAD
COMMON/FINK/ZM(NMAX),GAMI,RADDEG,DEGRAD,R90,R180,R270,R360,IOPSHP
COMMON/DARE/WEIGHT,XICG,AGRAB,FZP,VELF
COMMON/SHAP/DELS(70),S(70),R(70),THETA(70),THETAF,RB,EJ,
&RJ,RN,RC,RO,XFA,XFF,XLP,XCG,SN,THETAN
DIMENSION VALUE(1)
TINDX=TIND
Z=VALUE(2)
DY=VALUE(3)
DZ=VALUE(4)
VCG=SQRT(DY*DY+DZ*DZ)
ALF=ALPHA*DEGRAD
SALF=SIN(ALF)
SGAM=SIN(VALUE(5))
CGAM=COS(VALUE(5))
WIB=VALUE(6)
YB= CGAM*DY + SGAM*DZ
YC=-SGAM*DY + CGAM*DZ

```

PENCO2D
WES' TWO-DIMENSIONAL PENETRATION CODE

```

TDVZ=0.5*VALUE(6)/YC
SB=0.
SC=0.
SE=0.
EPS=VCG*SYNEFEE
JL=0
JLP1=1
DO 410 I=1,NEL
SI = S(I)
RI = R(I)
1 IF(SI.GT.S0(JLP1)) GO TO 2
JL=JL+1
JLP1=JLP1+1
GO TO 1
2 IF(JL.EQ.0) GO TO 3
RM=TDVZ*(S0(JL)-SI)**2
C2=(SI-XCG)*SALF
RC2=RCAV(I)**2
RI2=RI**2
3 FA = Z + CGAM*SI
Q = SGAM*RI
BTEST=MIN(FA+Q,FA-Q)
IF(BTEST) 10,50,50
50 IF(VALUE(5)-4.71238) 412,412,410
10 DA=ADA(I)
YBI = YB - W1B*SI
CATE = CAT(I)
SATE = SAT(I)
RCTHCG=RI*CGAM*CATE
32 FF=W1B*RI
TB=0.; TC=0.; TE=0.
DO 411 J=1,NEMOV2
SALP = XSALP(J)
CALP = XCALP(J)
208 ZRN=FA-Q*CALP
IF(ZRN.GE.0.) GO TO 411
DO 12 K=2,NMAX
IF(ZM(K)+ZRN) 12,13,13
12 CONTINUE
GO TO 411
***** CHECK FOR FREE SURFACE EFFECTS *****
13 ZDAMAG=FA-Q*CALP*(1.+SHRGD(K)*SATE)+SHRGD(K)*RCTHCG
DO 14 KDAM=K-1,K+1
IF(ZM(KDAM)+ZDAMAG) 14,15,15
14 CONTINUE
15 IF(K.EQ.KDAM) GO TO 19
IF(LMEDIA(K)) GO TO 17
IF(LMEDIA(KDAM)) GO TO 18
IF(YIELD(K).GT.YIELD(KDAM)) GO TO 18
GO TO 19
17 IF(.NOT.LMEDIA(KDAM)) GO TO 19
IF(SNUM(K).GT.SNUM(KDAM)) GO TO 19
18 K=KDAM
19 CONTINUE
***** END FREE SURFACE EFFECT CHECKS *****
B23=-CALP*SATE

```

PENC02D
WES' TWO-DIMENSIONAL PENETRATION CODE

```

QB=YBI
QC=YC - FF*CALP
ZC=B23*QB+CATE*QC
IF(ZC.LE.0.) GO TO 411
ZCC=YC*CATE + YB*B23
IF(ZCC.LT.0.) GO TO 411
***** CHECK FOR SEPARATION *****
IF(JL.EQ.0) GO TO 430
IF(ZCC.GE.EPS) GO TO 430
C1=-RI*CALP - C2 + RM
C3=SQRT(RC2 - RI2*SALP**2)
IF(CALP) 21,21,22
21 IF(C1.LT.C3) GO TO 411
GO TO 430
22 IF(-C1.LT.C3) GO TO 411
***** END CHECK FOR SEPARATION *****
430 ZB=SALP*QB
B21=-CALP*CATE
ZA=B21*QB-SATE*QC
SMALLV=SQRT(ZA**2+ZB**2+ZC**2)
VNV2=ZC/SMALLV
VNV=SQRT(VNV2)
IF(LMEDIA(K)) GO TO 451
SIGMAS=(HRD1(K)+SMALLV*HRD2(K))*VNV
GO TO 453
451 SIGMAS=(AKASER(I,K)+BKBSER(I,K)*SMALLV-ZRN*GK*CATE(K)*
& VNV)*VNV
453 AA=1.
BB=ARSIN(VNV2)
IF(BB.LT.ELAS) AA=SIN(BB*ALAS)
Y3=-AA*SIGMAS*CATE
TC=TC + Y3
IF(LOGCUT) GO TO 411
Y2=AA*SIGMAS*SATE*CALP
TB=TB + Y2
TE=TE - CALP*Y3
411 CONTINUE
TC=TC*DA
SC=SC+TC
IF(LOGCUT) GO TO 410
TB=TB*DA
SB=SB + TB
DATE=DA*RI*TE
SE=SE + (DATE - TB*SI)
410 CONTINUE
412 SC=2.*SC
SB=2.*SB
SE=2.*SE
FYP=CGAM*SB-SGAM*SC
FZP=SGAM*SB+CGAM*SC
RETURN
END

```

***** CALCULATE ATTACK ANGLE ALPHA *****

PENCO2D
WES' TWO-DIMENSIONAL PENETRATION CODE

```
SUBROUTINE CALF
PARAMETER NMAX=7, NMAX1=NMAX+1
COMMON/HOPE/VALUE(6)/BACK/ALPHA/FINK/DUMMY(NMAX1),RADDEG,
&DEGRAD,R90,R180,R270,R360
IF(VALUE(3)) 515,510,515
515 IF(VALUE(4)) 525,520,525
525 TH=ATAN(-(VALUE(3)/VALUE(4)))
GO TO 530
510 IF(VALUE(4).LT.0.) GO TO 512
ALPHA=VALUE(5)*RADDEG
GO TO 35
512 ALPHA=(VALUE(5)-R180)*RADDEG
GO TO 35
520 IF(VALUE(3).LT.0.) GO TO 522
ALPHA=(VALUE(5)-R270)*RADDEG
GO TO 35
522 ALPHA=(VALUE(5)-R90)*RADDEG
GO TO 35
530 IF(VALUE(4)) 531,520,533
533 IF(VALUE(3).LT.0.) GO TO 532
ALPHA=(VALUE(5)-TH-R360)*RADDEG
GO TO 35
532 ALPHA=(VALUE(5)-TH)*RADDEG
GO TO 35
531 ALPHA=(VALUE(5)-TH-R180)*RADDEG
35 IF(ALPHA-180.) 31,30,30
31 IF(ALPHA+180.) 32,32,34
32 ALPHA=ALPHA + 360.
GO TO 31
30 ALPHA=ALPHA - 360.
GO TO 35
34 RETURN
END
```

```
***** REVERSE GEOMETRY FOR TUMBLING PROJECTILE (ALPHA>90 DEG.) ***
SUBROUTINE BACKWARD
PARAMETER NMAX=7, NM3=NMAX+3
COMMON/TRANS/SNO,XCG0/HOPE/VALUE(6)/BACK/ALPHA/FINK/DUMMY(NM3),
&R90,R180/PXTRA/STEST,DL,ARATIO,NEL/BIGS/ADA(70),CAT(70),SAT(70)
COMMON/SHAP/DELS(70),S(70),R(70),THETA(70),DUM(10),XCG,SN
DATA ICNT/0/
DIMENSION SBCK(70),RBCK(70),DLBCK(70),THBCK(70)
ICNT=ICNT + 1
ITEST=(ICNT/2)*2
VALUE(5)=VALUE(5) - SIGN(1,VALUE(5)-R180)*R180
ALPHA=ALPHA - SIGN(1,ALPHA)*180
IF(ICNT.EQ.ITEST) GO TO 15
SN=0.001
XCG=-S(NEL) + 0.5*DELS(NEL)
GO TO 16
15 SN=SNO
XCG=XCG0
16 DO 10 J=1,NEL
JN=NEL - J + 1
```

PENCO2D
WES' TWO-DIMENSIONAL PENETRATION CODE

```

SBCK(J)=-S(JN)
RBCK(J)=R(JN)
DLBCK(J)=DELS(JN)
10 THBCK(J)=-THETA(JN)
DO 20 J=1,NEL
S(J)=SBCK(J)
R(J)=RBCK(J)
DELS(J)=DLBCK(J)
THETA(J)=THBCK(J)
ATEH=R90 - THETA(J)
CAT(J)=COS(ATEH)
SAT(J)=SIN(ATEH)
20 ADA(J)=DL*R(J)*DELS(J)/COS(THETA(J))
100 FORMAT(//,"PROBLEM RESTARTED -- PROJECTILE GOING BACKWARD",
&" NOW",//)
110 FORMAT(//,"PROBLEM RESTARTED -- PROJECTILE GOING FORWARD",
&" NOW",//)
CALL SEPARATE
IF(ICNT.EQ.1TEST) GO TO 35
WRITE(6,100)
GO TO 36
35 WRITE(6,110)
36 RETURN
END

```

```

***** READ AND ECHO-PRINT PROBLEM INPUT *****
SUBROUTINE READIN
PARAMETER NMAX=7
COMMON/TPLOT/TITLE,NLAY,NUMPLTS,ICHOOZ(8),ZCUT
COMMON/YOUNG/LMEDIA(NMAX),SNUM(NMAX),SHRGD(NMAX),DENSITY(NMAX)
&,YIELD(NMAX)/ROUND/ALAS,ELAS
COMMON/SEPAR/E/SYNEFEE,TANFEE,PHIMIN,RCAV(70),S0(10)/WT2/XIINV,XM
LOGICAL LMEDIA
DIMENSION XIRD(NMAX)
CHARACTER *1DDUM/1H /,TITLE*45
COMMON/PXTRA/STEST,PICON,ARATIO,NEL/INCR/DGAMAD,DALFAD,NUMSTP
COMMON/NEWT/FRAD,FANG,DTMAX,DTMIN,FREKOT,IREDUC,NUMNOS,NPRINT
COMMON/MSUN/ZSTOP,YSTOP,GAMSTP,VEL,ZSHIFT
COMMON/FINK/ZM(NMAX),GAMI,RADDEG,DEGRAD,R90,R180,R270,R360,IOPSHP
COMMON/SHAP/DELS(70),S(70),R(70),THETA(70),THETAF,RB,EJ,
&RJ,RN,RC,RO,XFA,XFF,XLP,XCG,SN,THETAN
COMMON/DARE/WEIGHT,XICG,AGRAB,FZP,VELF
COMMON/HOPE/VALUE(6),DER(6),THETAD(70),THETND,THETFD,TIMEF,
&TIMEI,GAMMAD,FREQI,NSTP,ALSTP/DOPE/WII,FYP,NEMOV2,ALPHAD
111 FORMAT(IH1)
1000 FORMAT(12I6)
1001 FORMAT(4E18.6)
1002 FORMAT(5E14.4)
998 FORMAT(3X,A45)
READ(5,998,END=999) TITLE
997 FORMAT(10X,A4)
WRITE(6,997) TITLE
WRITE(6,901)
901 FORMAT(//,3X,37(1H*),/,3X,

```

PENCO2D
WES' TWO-DIMENSIONAL PENETRATION CODE

```

* WES 2D NEWTONIAN PENETRATION CODE *,
2/,3X,37(1H*)
  READ(5,1000) NEMOV2,IOPSHP,NSTP,NLAY,NUMPLTS,NUMSTP,NUMNOS
  &,IREDUC,NPRINT
  WRITE(6,2000) NEMOV2,IOPSHP,NSTP,NLAY,NUMPLTS,NUMSTP,NUMNOS
  &,IREDUC,NPRINT
2000 FORMAT(/,3X,"NEMOV2=",I4,2X,"IOPSHP=",I4,2X,"NSTP=",I4,2X,
  &"NLAY=",I4,2X,"NUMPLTS=",I4,2X,"NUMSTP=",I4,2X,"NUMNOS=",I4,2X,
  &"IREDUC=",I4,2X,"NPRINT=",I4)
  IF(NUMPLTS.EQ.0) GO TO 950
  READ(5,1000)(ICHOOZ(I),I=1,NUMPLTS)
  WRITE(6,2002)(ICHOOZ(I),I=1,NUMPLTS)
2002 FORMAT(/,3X,"ADDITIONAL PLOTS...",7(I4))
950 READ(5,1001) DELS(1),ARATIO,ELAS,DGAMAD
  WRITE(6,1100) DELS(1),ARATIO,ELAS,DGAMAD
1100 FORMAT(/,3X,"DELS(1)=",E13.6,
  &3X,"ARATIO=",E13.6,3X,"ELAS=",E13.6,3X,"DGAMAD=",E13.6)
  READ(5, 1001) RB,EJ,RJ,RN
  WRITE(6,1003)RB,EJ,RJ,RN
1003 FORMAT(/,3X,3HRB=,E13.6,1X,3HEJ=,E13.6,1X,3HRJ=,E13.6,1X,3HRN=
  &E13.6)
  READ(5, 1001) RC,RO,XFA,XFF
  WRITE(6,1020)RC,RO,XFA,XFF
1020 FORMAT(/,3X,3HRC=,E13.6,1X,3HRO=,E13.6,1X,4HXFA=,
  &E13.6,1X,4HXFF=,E13.6)
  READ(5, 1001) XLP,XCG,SN,THETND
  WRITE(6,1004)XLP,XCG,SN,THETND
1004 FORMAT(/,3X,4HXL=,E13.6,1X,4HXC=,E13.6,1X,
  &3HSN=,E13.6,1X,7HTHETND=,E13.6)
  READ(5,1001) THETFD,VEL,FREKOT,DALFAD
  WRITE(6,1005) THETFD,VEL,FREKOT,DALFAD
1005 FORMAT(/,3X,"THETFD=",E13.6,1X,"VEL=",E13.6,1X,"FREKOT=",
  &E13.6,3X,"DALFAD=",E13.6)
  READ(5,1001) GAMSTP,YSTOP,ZSTOP,ZSHIFT
  READ(5,1001) FRAD,FANG,DTMIN,DTMAX
  READ(5,1001) WEIGHT,XICG,ALPHAD,ALSTP
  READ(5,1001) TIMEI,TIMEF,FREQI
  READ(5,1001) PHIMIN,VELF,WII,GAMMAD
  WRITE(6,1006) GAMSTP,YSTOP,ZSTOP,ZSHIFT
  WRITE(6,1007) FRAD,FANG,DTMIN,DTMAX
  WRITE(6,1009) WEIGHT,XICG,ALPHAD,ALSTP
  WRITE(6,1010) TIMEI,TIMEF,FREQI
  WRITE(6,1011) PHIMIN,VELF,WII,GAMMAD
1006 FORMAT(/,3X,7HGAMSTP=,E13.6,2X,6HYSTOP=,E13.6,2X,
  &6HZSTOP=,E13.6,2X,"ZSHIFT=",E13.6)
1007 FORMAT(/,3X,"FRAD=",E13.6,2X,"FANG=",
  &E13.6,2X,"DTMIN=",E13.6,2X,"DTMAX=",E13.6)
1009 FORMAT(/,3X,"WEIGHT=",E13.6,2X,"XICG=",E13.6,2X,"ALPHAD=",
  &E13.6,2X,"ALSTP=",E13.6)
1010 FORMAT(/,3X,"TIMEI=",E13.6,2X,"TIMEF=",
  &E13.6,2X,"FREQI=",E13.6)
1011 FORMAT(/3X,"PHIMIN=",E13.6," DEG.",2X,5HVELF=,E13.6,
  & 2X,4HW1I=,E13.6,2X,7HGAMMAD=,E13.6)
  READ(5,1002)(SNUM(I),DENSITY(I),YIELD(I),ZM(I),XIRD(I),I=2,NLAY)
  WRITE(6,1301)
1301 FORMAT(//,5X,"LAYER",12X,"SNUM",14X,"DENSITY",7X,

```

PENCO2D
WES' TWO-DIMENSIONAL PENETRATION CODE

```

"YIELD",10X,"ZM",16X,"XIRD",/,6X,"NO.",7X,
"&"(YOUNG'S S-NUMBER)",2X,"(LB-SEC12/INT4)",3X,
&"(PSI)",3X,"(LAYER DEPTH-IN.)",2X,"(TENS.RIG.INDEX)",/,,
&/,7X,"1 * * * A I R * * * N O R E S I S T A N C E * * *"
      WRITE(6,1302)(DDUM,I,SNUM(I),DENSITY(I),YIELD(I),
      & ZM(I),XIRD(I),I=2,NLAY)
1302 FORMAT(/,4X,A1,I3,12X,F6.2,10X,1PE12.4,2X,E11.3,E12.3,
      & 7X,0PF9.1)
      PHIMIN=PHIMIN*DEGRAD
      SYNEFEE=SIN(PHIMIN)
      TANFEE=TAN(PHIMIN)
      XIINV=AGRAV/XICG
      W=WEIGHT; X=0.
      XM = AGRAV / W
      DO 235 J=2,NLAY
      SHRGD(J)=XIRD(J)**(1./3.)
      IF(SNUM(J)) 994,991,992
991  LMEDIA(J)=.F.
      GO TO 235
992  LMEDIA(J)=.T.
235  CONTINUE
      ZCUT=-MAX(XLP-XCG,RO*(SHRGD(NLAY)+1))-RO
      ZCUT=ZCUT-ZM(NLAY-1)
      IF(IOPSHP.NE.0) GO TO 14
      READ(5,1000) NEL
      WRITE(6,1150) NEL
1150 FORMAT(//,3X,"NEL (NO. OF ELEMENTS) =",I4)
      WRITE(6,1151)
1151 FORMAT(//,3X,"THETA-ARRAY...")
      READ(5,1001) (THETAD(I),I=1,NEL)
      WRITE(6,1001) (THETAD(I),I=1,NEL)
      WRITE(6,1152)
1152 FORMAT(//,3X,"R-ARRAY...")
      READ(5,1001) (      R(I),I=1,NEL)
      WRITE(6,1001) (      R(I),I=1,NEL)
      WRITE(6,1153)
1153 FORMAT(//,3X,"DELS-ARRAY...")
      READ(5,1001) (      DELS(I),I=1,NEL)
      WRITE(6,1001) (      DELS(I),I=1,NEL)
14    CONTINUE
      RETURN
994  WRITE(6,995)
995 FORMAT(//,2X,"NEGATIVE S-NUMBER...RUN ABORTED.")
999 CALL CALCMP(0.,0.,9999,2)
      CALL DETACH(5,,)
      STOP
      END

```

```

***** CONVERT CALCOMP CRT PLOT CALL TO CALCOMP DRUM PLOT CALL ****
SUBROUTINE CALCMP(X,Y,IZ,I)
DATA J0/-1/
IF(I.EQ.2) GO TO 30
IF(IZ.LT.2) GO TO 10
CALL PLOT(X,Y,2)

```

PENCO2D
WES' TWO-DIMENSIONAL PENETRATION CODE

```
10 GO TO 20
10 CALL PLOT(X,Y,3)
GO TO 20
30 IF(IZ.EQ.9999) GO TO 40
J0=-J0
CALL PLOT(-2.,-2.5,-3)
CALL PLOT(0.,11.,2)
CALL PLOT(8.5,11.,2)
CALL PLOT(8.5,0.,2)
CALL PLOT(0.,0.,2)
IF(J0.EQ.-1) GO TO 22
CALL PLOT(2.,14.5,-3)
GO TO 20
22 CALL PLOT(12.,-9.5,-3)
GO TO 20
40 CALL PLOT(10.,0.,3)
CALL PLOT(0.,0.,999)
20 RETURN
END
```

APPENDIX H

NOTATION

dA	Differential area on projectile surface
F_y, F_z	Net force components in the projectile-fixed y- (lateral) and z- (axial) directions, respectively
F_Y, F_Z	Net force components in the target-fixed Y- and Z- directions, respectively
I	Projectile's transverse mass moment of inertia about its CG
I_T	Free-surface parameter
λ	Logic flag used to signify that the end of the problem is approaching
L_N	Projectile nose length
m	Projectile mass
M	Total moment exerted on the projectile about its CG
\vec{n}	Unit vector normal to projectile's surface
N	Young's nose performance coefficient (Reference 2)
N_D	Number of integration time steps required to pass through the thinnest target layer
r_D	Damage radius
r_o	Projectile aftbody radius
r_p	Local cylindrical radius at a given point on the projectile surface
$s(t)$	Path length traveled by projectile up to time t
S	Young's S-number (Reference 2), soil penetrability index
t	Time
t_f	Maximum time allowed for current problem
t_o	Initial time
T_{\min}	Thickness of the thinnest target layer in a multilayer problem
v	Velocity of any point on projectile surface
v_n	Outward normal component of local velocity

v	Projectile CG velocity
v_f	Minimum CG velocity allowed for current problem
v_o	Projectile impact velocity
w	Projectile weight
x_{CG}	Distance from projectile nose tip to CG
y, z	Lateral and axial position in projectile, respectively
Y_{stp}	Maximum absolute value of Y allowed for current problem
Y, Z	CG displacement components in the target-fixed Y - and Z -directions
y_o, \dot{y}_o	Initial value of horizontal position and velocity of CG
\dot{y}, \dot{z}	Velocity components of CG in the target-fixed Y - and Z -directions
\ddot{y}, \ddot{z}	Net acceleration components in the target-fixed Y - and Z -directions, respectively
$\dot{y}_{ave}, \dot{z}_{ave}$	Average values of \dot{y} and \dot{z} during current time step
\dot{y}_p, \dot{z}_p	Values of \dot{y} and \dot{z} at the beginning of current time step
z_f	Total path length traveled by a penetrator from impact to rest
z_o, \dot{z}_o	Initial values of vertical position and velocity of CG
z_s	Vertical shift in initial projectile position, positive downwards
Z_{stp}	Minimum value of Z allowed for current problem
Δt	Actual integration time-step size
$\Delta t'$	Time step calculated using translational parameters
$\Delta t''$	Time step calculated using rotational parameters
Δv	Change in CG velocity during time step Δt
$\Delta Y, \Delta Z$	Change in CG displacement components in the target-fixed Y - and Z -directions during time step Δt
$\Delta \dot{y}, \Delta \dot{z}$	Change in CG velocity components in the target-fixed Y - and Z -directions during time step Δt
$\Delta \gamma$	Change in projectile orientation during time step Δt
$\Delta \dot{\gamma}$	Change in angular velocity during time step Δt

α	Projectile angle of attack (i.e., angle between projectile axis and CG velocity vector, positive when velocity vector is clockwise from axis)
α_0	Initial value of angle of attack α
α_{stp}	Maximum absolute value of α allowed for current problem
γ	Obliquity angle, measured counterclockwise from target-fixed Z-axis to projectile-fixed z-axis
$\gamma_0, \dot{\gamma}_0$	Initial values of obliquity and angular velocity
γ_{stp}	Maximum value of γ allowed for current problem
$\dot{\gamma}$	Angular velocity
$\dot{\gamma}_{ave}$	Average value of $\dot{\gamma}$ during current time step
$\dot{\gamma}_p$	Value of $\dot{\gamma}$ at the beginning of current time step
$\ddot{\gamma}$	Angular acceleration
ξ	Angle between local tangential velocity component and local total velocity vector
ρ	Mass density of either a hard target material or a projectile structure material
σ	Local compressive normal stress on projectile surface
σ^*	Smoothed normal stress function (Appendix E)

DISTRIBUTION LIST

<u>Address</u>	<u>No. of Copies</u>
DOD	
Director, Defense Advanced Research Project Agency, 1400 Wilson Blvd., Arlington, Virginia 22209 ATTN: Technical Library	1
Defense Documentation Center, Cameron Station, Alexandria, Virginia 22314 ATTN: TC/Myer B. Kahn	12
Director, Defense Intelligence Agency, Washington, D. C. 20301 ATTN: Technical Library	1
Defense Mapping Agency, Washington, D. C. 20305 ATTN: H. Lindsey	1
Director, Defense Nuclear Agency, Washington, D. C. 20305 ATTN: STSI (Archives) STTL (Technical Library) SPSS DDST TISI	1 2 15 1 2
Under Secretary of Defense for Research and Engineering, Department of Defense, Washington, D. C. 20301 ATTN: Deputy Director, Tactical Warfare	1
Commander, Field Command, Defense Nuclear Agency, Kirtland Air Force Base, New Mexico 87115 ATTN: FCTMD	1
Interservice Nuclear Weapons School, Kirtland Air Force Base, New Mexico 87115 ATTN: Technical Library	1
Director, Joint Strategic Target Planning Staff, JCS, Offutt Air Force Base, Nebraska 68113 ATTN: Science and Technology Info Library	1

<u>Address</u>	<u>No. of Copies</u>
<u>Army</u>	
Assistant Chief of Staff for Force Development, Department of the Army, Washington, D. C. 20310 ATTN: Technical Library	1
Commander, U. S. Army Corps of Engineers, Washington, D. C. 20314 ATTN: DAEN-ASI-L DAEN-MPE-T DAEN-RDL DAEN-RDM	2 2 1 1
Chief of Staff for Research, Development, and Acquisition, Department of the Army, Washington, D. C. 20310 ATTN: Technical Library	1
Commander, Harry Diamond Laboratories, Department of the Army, 2800 Powder Mill Road, Adelphi, Maryland 20783 ATTN: AMXDO-RBH/J. Gwaltney	1
Commanding Officer, Picatinny Arsenal, Dover, New Jersey 07801 ATTN: Technical Library	1
Director, U. S. Army Ballistic Research Laboratories, Aberdeen Proving Ground, Maryland 21005 ATTN: G. Roecker Technical Library/Edward Baicy	1 1
Division Engineer, U. S. Army Engineer Division, Huntsville, P. O. Box 1600, West Station, Huntsville, Alabama 35807 ATTN: HNDED-R/Michael M. Dembo	1
Commander, Army Materials and Mechanics Research Center, Watertown, Massachusetts 02172 ATTN: Technical Library	1
Commander, U. S. Army Material Development and Readiness Command, 5001 Eisenhower Avenue, Alexandria, Virginia 22333 ATTN: Technical Library	1
Commander, U. S. Army Missile Command, Redstone Arsenal, Alabama 35809 ATTN: Technical Library DRCPM-PE-X (William K. Jann)	1 1

<u>Address</u>	<u>No. of Copies</u>
<u>Army (Continued)</u>	
Commander, U. S. Army Nuclear and Chemical Agency, 7500 Backlick Road, Bldg 2073, Springfield, Virginia 22150	
ATTN: Technical Library	1
Commander, U. S. Army Armament Command, Rock Island Arsenal, Rock Island, Illinois 61201	
ATTN: Technical Library	1
<u>Navy</u>	
Chief of Naval Research, Department of the Navy, Arlington, Virginia 22217	
ATTN: Technical Library	1
Naval Civil Engineering Laboratory, Port Hueneme, California 93043	
ATTN: Technical Library	1
Naval Facilities Engineering Command, 200 Stovall Street, Alexandria, Virginia 22332	
ATTN: Technical Library	1
Director, Naval Research Laboratory, Washington, D. C. 20375	
ATTN: Code 2027/Technical Library	1
Commander, Naval Surface Weapons Center, White Oak, Silver Spring, Maryland 20910	
ATTN: Technical Library, Code 730	1
Commander, Naval Surface Weapons Center, Dahlgren Laboratory, Dahlgren, Virginia 22448	
ATTN: Technical Library	1
Commander, Naval Weapons Center, China Lake, California 93555	
ATTN: Code 2663, Dr. Carl Austin	1
Commander, Naval Weapons Evaluation Facility, Kirtland Air Force Base, Albuquerque, New Mexico 87117	
ATTN: Technical Library	1

<u>Address</u>	<u>No. of Copies</u>
<u>Air Force</u>	
Air Force Armament Laboratory (AFSC), Eglin Air Force Base, Florida 32542	
ATTN: DLYV (Mr. John Collins)	1
DLJW (Mr. G. R. Griner)	1
Technical Library	1
Air Force Weapons Laboratory (AFSC), Kirtland Air Force Base, New Mexico 87117	
ATTN: Technical Library	1
Headquarters, Air Force Systems Command, Andrews Air Force Base, Washington, D. C. 20331	
ATTN: Technical Library	1
Oklahoma State University Field Office, P. O. Box 1925, Eglin Air Force Base, Florida 32542	
ATTN: Ed Jackett	1
Air Force Institute of Technology, Air University, Wright-Patterson Air Force Base, Ohio 45433	
ATTN: Library	1
<u>Other Government Agencies</u>	
Lawrence Livermore National Laboratory, P. O. Box 808, Livermore, California 94550	
ATTN: Technical Library	1
Nuclear Regulatory Commission, Directorate of Licensing Regulations, Washington, D. C. 20555	
ATTN: Mr. Lawrence Shao, Chief, Struc Engrg Br	1
Sandia National Laboratories, P. O. Box 5800, Albuquerque, New Mexico 87115	
ATTN: Mr. William Patterson	1
C. Wayne Young	1
<u>DOD Contractors</u>	
Avco Corporation, 201 Lowell Street, Wilmington, Massachusetts 01887	
ATTN: David Henderson, Government Products Group	1
Research Library A830, Room 7201	1

<u>Address</u>	<u>No. of Copies</u>
<u>DOD Contractors (Continued)</u>	
Boeing Aerospace Company, P. O. Box 3999, Mail Stop 13-11, Seattle, Washington 98124 ATTN: Aerospace Library	1
Braddock, Dunn, & McDonald, Inc., 1920 Aline Avenue, Vienna, Virginia 22180 ATTN: Technical Library	1
California Research and Technology, Inc., 6269 Variel Avenue, Woodland Hills, California 91367 ATTN: Dr. K. N. Kreyenhagen	1
Institute for Defense Analyses, 400 Army-Navy Drive, Arlington, Virginia 22202 ATTN: IDA Librarian Ruth S. Smith	1
Lockheed Missiles and Space Company, Inc., 3251 Hanover Street, Palo Alto, California 94304 ATTN: Tech Info Center D/COLL	1
Martin Marietta Aerospace, Orlando Division, P. O. Box 5837, Orlando, Florida 32805 ATTN: M. Anthony	1
New Mexico Engineering Research Institute, University of New Mexico, Box 25, University Station, Albuquerque, New Mexico 87131 ATTN: G. E. Triandafalidis	1
Pacifica Technology, P. O. Box 148, Del Mar, California 92014 ATTN: R. L. Bjork	1
Physics International Company, 2700 Merced Street, San Leandro, California 94577 ATTN: Doc Control for Mr. Larry Behrman	1
R&D Associates, P. O. Box 9695, Marina del Rey, California 90291 ATTN: Technical Library	1
TRW Defense & Space Systems Group, Mail Station R1/2154, One Space Park, Redondo Beach, California 90278 ATTN: Technical Info Center/S-1930	1

In accordance with letter from DAEN-RDC, DAEN-ASI dated 22 July 1977, Subject: Facsimile Catalog Cards for Laboratory Technical Publications, a facsimile catalog card in Library of Congress MARC format is reproduced below.

Creighton, Daniel C.

Non-normal projectile penetration in soil and rock : User's guide for computer code PENCO2D / by Daniel C. Creighton (Structures Laboratory, U.S. Army Engineer Waterways Experiment Station). -- Vicksburg, Miss. : The Station ; Springfield, Va. ; available from NTIS, 1982.

138 p. : ill. ; 27 cm. -- (Technical report ; SL-82-7)

Cover title.

"September 1982."

Final report.

"Prepared for Defense Nuclear Agency under DNA Task Y99QAXSC, Work Unit 00042, "Earth Penetrator Support."

Bibliography: p. 68.

1. Soil penetration test. 2. Projectiles.
3. MOMENT (Computer program). 4. PENCO2D (Computer program). 5. Soils--Testing. 1. United States.

Creighton, Daniel C.

Non-normal projectile penetration in soil and : ... 1982.
(Card 2)

Defense Nuclear Agency. II. U.S. Army Engineer Waterways Experiment Station. Structures Laboratory.
III. Title IV. Series: Technical report (U.S. Army Engineer Waterways Experiment Station) ; SL-82-7.
TA7.W34 no.SL-82-7